Freie Universität Berlin
Department of Mathematics and Computer Science

Numerical Investigations of Reduced Order Models for Convection-Diffusion Equations

Shoucheng Ling

Master Thesis

supervised by
Prof. Dr. Volker John
Dr. Alfonso Caiazzo

Berlin, February 16, 2016

Declaration of Originality

I declare that this thesis is the result of my independent work. All sources and auxiliary materials used in this thesis are cited completely.

Berlin, February 16, 2016.

Contents

1 Introduction

At present, numerical simulations of complex partial differential equations for modern real-life process are required. The process of simulating these equations by the well known numerical discretization methods, e.g., finite difference method (FDM), finite element method (FEM) and finite volume method (FVM), becomes more and more time and computation consuming, due to the complexity of the problems and the need of large storage of data. For this reason, finding methods that can significantly reduce the computation become a popular issue. Reduce Order Modeling (ROM) is such a method that approximate the original problems in much lower-dimensional spaces but still with acceptable accuracy. The Proper Orthogonal Decomposition (POD) method is one of the most popular ROM methods. It extracts basis functions from a set of data, and approximates these data by the computed basis functions with much less degrees of freedom. In this thesis, POD will be exclusively considered.

The convection-diffusion-reaction equations model the concentration of chemical species in the reaction, diffusion and convection processes. It is well known that the Galerkin finite element method (G-FEM) fails to solve the convection-dominated problems, due to the sharp layers of the solution, and hence the Streamline-Upwind Petrov-Galerkin (SUPG) method, one of the most popular stabilization discretization methods proposed in [2], is applied.

Numerical research will focus on the study of the factors which may have impact on the ROMs, e.g., the amount of snapshots taken from the finite element methods, the rank of the extracted basis functions and different inner products for the computation of basis functions, etc.

This thesis will be organized as follow. In Chapter 2, the derivation of POD method and the computation of basis functions will be given as a review. In Chapter 3, the convection-dominated convection-diffusion-reaction equations are introduced, and some properties of these equations will be presented. Moreover, the time and space discretization method applying on this type of equation will be given.

The numerical results will be shown in Chapter 4, in which five examples are studied. The first one investigate the differences of applying G-FEM and SUPG-FEM on a relatively simple convection-dominated equation and of using different inner products for POD modes and ROMs.

The second and third examples form a comparison of using POD method for non-convection-dominated problems and convection-dominated problems, and the fourth example mainly investigate the impact of using two different inner products on the POD modes and ROMs for a more complex convetion-dominated problem. The last example fetches the snapshots from the analytical solution, and to see if this helps to get more accurate results compared to the case by taking snapshots from SUPG-FEM.

Finally, a summary of the results and an outlook will be given in Chapter 5.

2 Proper orthogonal decomposition method

The Proper Orthogonal Decomposition (POD) method is one of the most popular techniques for the Reduced Order Modeling (ROM). It has been used in many areas like signal analysis, fluid dynamics, image processing, and data compression, etc. The basic idea of this method is to extract basis functions from a given set of data based on the least squares method. This set of basis functions represents the given data in a good way and therefore can be used for simulations where one expects to get similar data but using much less degrees of freedom. The use of these basis functions in order to run cheap simulations is called Reduced Order Modeling. Since the number of these basis functions is usually much smaller than the dimension of the original data, which usually could be obtained by solving the partial differential equations using the finite element method or from experimental data. One can approximate the solutions in the new simulations with much fewer number of degrees of freedom. By using the POD method, the reduced order model with best approximation accuracy and very few degrees of freedom can be achieved.

2.1 The space-continuous case

For given $n \in \mathbb{N}$, let

$$0 = t_0 < t_1 < \cdots < t_n = T$$

denote the discrete grid of time interval $[0, T]$, and $\Delta t_i = t_i - t_{i-1}, i = 1, \ldots, n$. For the sake of simplicity, it is assumed that the snapshots are computed at equidistant time steps using the finite element method, i.e., $\Delta t_1 = \cdots = \Delta t_n$. Consider a function $\mathbf{u}(t, \mathbf{x}) : [0, T] \times \Omega \to \mathbb{R}^d$, take $u_1, , u_2, \ldots, u_M$ as discrete allocation of $u(t, \mathbf{x})$ at discrete times $t_1, t_2, \ldots, t_M$ in $[0, T]$ (so-called 'snapshots'), i.e.,

$$u_i = u_i(\mathbf{x}) = u(t_i, \mathbf{x}), (t_i, \mathbf{x}) \in [0, T] \times \Omega$$

and define

$$V^h = span\{u_1, u_2, \ldots, u_M\}.$$

V^h is called the ensemble of snapshots $\{u_i\}$, and assume $\dim V^h = D$, and it is assumed that at least one of the snapshots is nonzero.
The goal of the POD method is to find the basis functions $\{\varphi_l(\mathbf{x}) : \Omega \to \mathbb{R}^d\}_{l=1}^L$, for any $L \in \{1, \ldots, D\}$, that represent the original snapshot in the best possible way. These basis functions span a subspace of the original space of the snapshots V^h. In this subspace, the mean square error between the elements u_i and its orthogonal projection, which is represented by the basis functions $\{\varphi_l(\mathbf{x})\}_{l=1}^L$, is minimized, which gives the best approximation

$$\underset{\varphi_1, \ldots, \varphi_L}{\arg\min} \sum_{m=1}^{M} \tau \left|\left| u(t_m, \mathbf{x}) - \sum_{l=1}^{L} \alpha_l(t_m) \varphi_l(\mathbf{x}) \right|\right|^2 \tag{2.1}$$

subject to

$$(\varphi_i, \varphi_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \tag{2.2}$$

where τ is the equidistant time step, and $\{\alpha_l(t) : [0,T] \to \mathbb{R}\}_{l=1}^{L}$ are the coefficients with respect to time.
Note that (2.1) is the composite trapezoidal approximation with modification of the integral

$$\underset{\varphi_1,\ldots,\varphi_L}{\arg\min} \int_0^T \left\| u(t_m, \mathbf{x}) - \sum_{l=1}^{L} \alpha_l(t_m)\, \varphi_l(\mathbf{x}) \right\|^2 .$$

It is assumed that the norm is induced by suitable inner product $(\cdot,\cdot)$, and the basis functions $\{\varphi_l\}_{l=1}^{L}$ are orthonormal with respect to this inner product.

For $L = D$, $\{\varphi_l\}_{l=1}^{D}$ is an orthonormal basis of V^h. From Hilbert space theory and Parseval's identity, every element u_m can be represented as

$$u_m = \sum_{l=1}^{D} (u_m, \varphi_l)\, \varphi_l. \tag{2.3}$$

Hence we can represent $\alpha_l(t_m)$ as

$$\alpha_l(t_m) = (u_m, \varphi_l). \tag{2.4}$$

Inserting expression (2.4) into (2.1), and using the orthonormality of $\{\varphi_l\}_{l=1}^{L}$, problem (2.1) can be reformulated as

$$\begin{aligned} &\underset{\varphi_1,\ldots,\varphi_L}{\arg\min} \sum_{m=1}^{M} \tau \left(u_m - \sum_{l=1}^{L} (u_m, \varphi_l)\, \varphi_l, u_m - \sum_{l=1}^{L} (u_m, \varphi_l)\, \varphi_l \right) \\ &= \underset{\varphi_1,\ldots,\varphi_L}{\arg\min} \sum_{m=1}^{M} \tau \left[\|u_m\|^2 - \left(2u_m \sum_{l=1}^{L} (u_m, \varphi_l)\, \varphi_l \right) + \sum_{l=1}^{L} (u_m, \varphi_l)^2 \right] \\ &= \underset{\varphi_1,\ldots,\varphi_L}{\arg\min} \sum_{m=1}^{M} \tau \left[\|u_m\|^2 - \sum_{l=1}^{L} (u_m, \varphi_l)^2 \right]. \end{aligned} \tag{2.5}$$

Since τ is a constant, the result of (2.5) does not depend on τ. For simplifying, without loss of generality, it is assumed that $\tau = 1$.
Since the first term of (2.5) is a certain value which is independent of φ_l, problem (2.1) is equivalent to maximizing the second term. To solve this optimization problem, the method of Lagrange multipliers is considered as a useful strategy to find the local maxima of the function

$$\sum_{m=1}^{M} \sum_{l=1}^{L} (u_m, \varphi_l)^2 .$$

The Lagrange functional is defined as

$$\Lambda(\varphi_1, \ldots, \varphi_L; \lambda_1, \ldots, \lambda_L) = \sum_{m=1}^{M} \sum_{l=1}^{L} (u_m, \varphi_l)^2 - \sum_{l=1}^{L} \lambda_l [(\varphi_l, \varphi_l) - 1].$$

The maxima of $\sum_{m=1}^{M}\sum_{l=1}^{L}(u_m,\varphi_l)^2$ for the original constrained problem must be obtained at the stationary points, i.e., it holds

$$\begin{cases}\partial_{\varphi_l}\Lambda = 0,\\ \partial_{\lambda_l}\Lambda = 0.\end{cases} \tag{2.6}$$

The first condition of Eq.(2.6) is equivalent to

$$2\sum_{m=1}^{M}(u_m,\varphi_l)\cdot\frac{\partial(u_m,\varphi_l)}{\partial\varphi_l} - 2\lambda_l\frac{\partial(\varphi_l,\varphi_l)}{\partial\varphi_l} = 0.$$

The term $\frac{\partial(u_m,\varphi_l)}{\partial\varphi_l}$ is the functional derivative of $F=(u_m,\varphi_l)$ with respect to the functions φ_l. To calculate it, the definition of the functional derivative is introduced.

Definition 2.1.1. *Given a manifold M representing functions ρ and a functional F defined as*
$F:\ M\to\mathbb{R}$ or $F:\ M\to\mathbb{C}$. The functional derivative is defined by

$$\begin{aligned}\int\frac{\delta F}{\delta\rho}(x)\phi(x)dx &= \lim_{\varepsilon\to 0}\frac{F(\rho+\varepsilon\phi)-F(\rho)}{\varepsilon}\\ &= \left[\frac{d}{d\varepsilon}F(\rho+\varepsilon\phi)\right]_{\varepsilon=0},\end{aligned}$$

where ϕ is an arbitrary function, and the quantity $\varepsilon\phi$ is called the variation of ρ.

For more details about the definition of the functional derivative, see [17].

Using the definition of functional analysis, one gets

$$\frac{\partial(u_m,\varphi_l)}{\partial\varphi_l} = (u_m,v),\quad \forall v\in V^h.$$

By the similar calculation, it holds

$$\frac{\partial(\varphi_l,\varphi_l)}{\partial\varphi_l} = (\varphi_l,v),\quad \forall v\in V^h.$$

Since v is independent of t_m, the first condition of Eq.(2.6) can be reformulated as follows

$$\sum_{m=1}^{M}(u_m,\varphi_l)(u_m,v) = \left(\sum_{m=1}^{M}(u_m,\varphi)u_m,v\right) = \lambda_l(\varphi_l,v)$$

for all $v\in V^h$. It is equivalent to

$$\sum_{m=1}^{M}(u_m,\varphi_l)u_m = \lambda_l\varphi_l,\quad l=1,\dots,L. \tag{2.7}$$

For more details, see, e.g., [3][22].

2.2 The space-discrete case

It is assumed that the snapshots are finite element solutions of partial differential equations, hence the function $u(t_m, \mathbf{x})$ can be written in discrete form by a finite number of degrees of freedom. For convenience, in this section and also in the following chapter, the dimension of the snapshot space will still be denoted as D, which is in general different from the dimension D defined in the continuous case above. Let $\{\mathbf{x}_n\}_{n=1}^N$ be the spatial nodes of the finite element method, $\{\psi_n(\mathbf{x})\}_{n=1}^N$ be the nodal basis with $\psi_n(\mathbf{x}_k) = \delta_{nk}$. Then we can represent u_i and φ_l in the form

$$\begin{aligned} u_i &= \sum_{n=1}^{N} u(t_i, \mathbf{x}_n)\, \psi_n(\mathbf{x}), \\ \varphi_l &= \sum_{n=1}^{N} \varphi_l(\mathbf{x}_n)\, \psi_n(\mathbf{x}). \end{aligned} \tag{2.8}$$

By the practical requirements, it will be assumed that $L \leqslant D$ and $M \ll N$.
Inserting (2.8) into (2.7) gives

$$\begin{aligned} &\sum_{m=1}^{M} (u_m, \varphi_l)\, u_m \\ =&\sum_{m=1}^{M} \left(\underline{u_m}^T S \underline{\varphi_l}\right) \underline{u_m}^T \vec{\psi} \\ =&U^T U S \underline{\varphi_l} \vec{\psi} \\ =&\lambda_l \underline{\varphi_l} \vec{\psi} \end{aligned} \tag{2.9}$$

where $\vec{\psi} = (\psi_1, \cdots, \psi_N)$ denotes the vector form of the local basis. For a function $u_m \in V^h$, $\underline{u_m}$ denotes the vector of its coefficients with respect to the basis $\{\psi_n\}_{n=1}^N$, and the same for $\underline{\varphi_l}$. The matrix $S \in \mathbb{R}^{N\times N}$ with $S_{kn} = (\psi_n, \psi_k)$ is obviously symmetric and positive definite, and the matrix U is used to collect the snapshot data of $u(t, \mathbf{x})$ on the nodes $\{\mathbf{x}_n\}_{n=1}^N$ at time $t_1, \ldots, t_M$

$$U = \begin{pmatrix} u_1^1 & u_2^1 & \cdots & u_M^1 \\ u_1^2 & u_2^2 & \cdots & u_M^2 \\ \vdots & \vdots & \vdots & \vdots \\ u_1^N & u_2^N & \cdots & u_M^N \end{pmatrix}.$$

The matrix $U \in \mathbb{R}^{N\times M}$ is called snapshot matrix with $(U)_{nm} = u(t_m, \mathbf{x}_n) = u_m^n$.
From the calculation of (2.9), the discrete eigenvalue problem can be obtained as

$$UU^T S \underline{\varphi_l} = \lambda_l \underline{\varphi_l}, \quad l = 1, \ldots, L. \tag{2.10}$$

Multiplying both sides of (2.10) from left by $S^{1/2}$

$$S^{1/2} UU^T S \underline{\varphi_l} = \lambda_l S^{1/2} \underline{\varphi_l},$$

by the symmetry of S, it can be easily seen that the original eigenvalue problem (2.10) can be reformulated as a new eigenvalue problem with the matrix $S^{1/2}UU^TS^{T/2}$ since

$$S^{1/2} UU^T S^{T/2} S^{1/2} \underline{\varphi_l} = \lambda_l S^{1/2} \underline{\varphi_l}.$$

Note that $S^{1/2}UU^TS^{T/2} = (S^{1/2}U)(S^{1/2}U)^T \in \mathbb{R}^{N\times N}$ is a symmetric matrix satisfying

$$\underline{z}^T S^{1/2}UU^TS^{T/2}\underline{z} = ((S^{1/2}U)^T\underline{z})^T((S^{1/2}U)^T\underline{z}) = \|(S^{1/2}U)^T\underline{z}\|_2^2 \geqslant 0, \quad \forall \underline{z} \in \mathbb{R}^N.$$

where $\|\cdot\|_2$ is the Euclidean norm. Hence it is positive semi-definite. By the properties of the semi-definite matrix, it is known that all eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_L$ are real and non-negative.

In practice, solving the eigenvalue problem (2.10) is in general computationally expensive since $N = dim(UU^TS)$ is usually very large, and also this matrix is quite dense. It is known that the eigenvalues of UU^TS are precisely the real numbers λ that satisfy the characteristic equation

$$\det(UU^TS - \lambda\mathbb{I}_N) = 0.$$

By the property of the determinant and Sylvester's determinant theorem, it holds

$$\begin{aligned}\det(UU^TS - \mathbb{I}_N) =& \lambda^N \det(\frac{1}{\lambda}U^TSU - \mathbb{I}_M)\\ =& \lambda^{N-M}\det(U^TSU - \lambda\mathbb{I}_M).\end{aligned}$$

Hence the characteristic function of the matrix $U^TSU \in \mathbb{R}^{M\times M}$ is the same as the matrix UU^TS, which implies that these two matrices have same eigenvalues. It is more economical to solve the eigenvalue problem with matrix U^TSU instead of UU^TS as normally $M \ll N$.

To find the relation between the orthonormal eigenvectors of the two matrices U^TSU and UU^TS, the method of singular value decomposition(SVD) is considered.

Theorem 2.2.1. *(The singular value decomposition (SVD)) If A is a real m-by-n matrix, and assume the rank of A is r, then there exist orthogonal matrices*

$$W = [\underline{w_1}, \ldots, \underline{w_m}] \in \mathbb{R}^{m\times m} \quad \textit{and} \quad V = [\underline{v_1}, \ldots, \underline{v_n}] \in \mathbb{R}^{n\times n}$$

such that

$$A = W\Sigma V^T \in \mathbb{R}^{m\times n}, \quad p = min\{m, n\}$$

where

$$\Sigma = \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix}$$

$S = diag(\sigma_1, \ldots, \sigma_r)$, $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r \geq \sigma_{r+1} = \cdots = \sigma_p = 0$, *where* $\{\sigma_i\}$ *are the square roots of the eigenvlues of* A^TA.

Proof. The proof can be found in [15], pp 164-176. □

For the SVD method, $\sigma_i = \sqrt{\lambda_i}$ are called the singular values of A, where λ_i are the eigenvalues (which are non-negative) of A^TA. And the subscript r is the index of the smallest positive eigenvalue of A^TA. The columns of W and V are called the left and right singular vectors of A corresponding to the eigenvalues $\sigma_1, \sigma_2, \ldots, \sigma_p$, respectively. For more details about this method, it is referred to [15][16].

Now consider again the matrix UU^TS. By applying the SVD method, it holds

$$S^{1/2}U = R\begin{pmatrix} G & 0 \\ 0 & 0 \end{pmatrix}K^T$$

where $G = diag\{\sigma_1, \ldots, \sigma_D\} \in \mathbb{R}^{D\times D}$ is a diagonal matrix with decreasing positive singular values $\{\sigma_i\}, \quad i = 1, \ldots, D$, and $R \in \mathbb{R}^{N\times N}$ with columns $\{\underline{r_i}\}_{i=1}^{N}$ and $K \in \mathbb{R}^{M\times M}$ with columns $\{\underline{k_i}\}_{i=1}^{M}$ are orthogonal matrices which consist of orthogonal eigenvectors of $S^{1/2}UU^TS^{T/2}$ and U^TSU corresponding to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$, respectively. These singular values $\{\sigma_i\}$ are the square roots of non-zero eigenvalues of both UU^TS and U^TSU. Moreover, it holds

$$S^{T/2}Uk_i = \sigma_i r_i, \quad U^TS^{1/2}r_i = \sigma_i k_i \quad for \quad i = 1, \ldots, D. \tag{2.11}$$

On one hand, the SVD method guarantees the existences of the singular values $\{\sigma_i\}$. On the other hand, it shows that the solutions $\varphi_1, \varphi_2, \ldots, \varphi_L$ of the optimization problem are given by the first L columns of the matrix K, since

$$U^TSU = K \begin{pmatrix} G & 0 \\ 0 & 0 \end{pmatrix} R^TR \begin{pmatrix} G & 0 \\ 0 & 0 \end{pmatrix} K^T = K \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix} K^T$$

$\Rightarrow$

$$S^{1/2}UU^TS^{T/2}K = K \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix}$$

where $Q = diag\{\lambda_1, \ldots, \lambda_D\}$.
In some literatures (e.g., [4]), the eigenvalues of the matrix U^TSU are referred to be "energy", and the corresponding eigenvectors are called as proper orthogonal modes. Consider the Frobenius norm of $S^{T/2}U$, which is defined as

$$\|S^{T/2}U\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}^2}, \tag{2.12}$$

where x_{ij} represents the element of matrix $S^{T/2}U$ at the i-th row and j-th column. One can regard the Forbenius norm as an indication of the "information" contained by the matrix $S^{T/2}U$. Furthermore, from the properties of the trace of the square matrix, one gets the relations

$$\begin{aligned} tr(U^TSU) &= \sum_i \lambda_i \\ tr(U^TSU) &= tr(S^{T/2}UU^TS^{1/2}) = \sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}^2. \end{aligned} \tag{2.13}$$

Inserting (2.13) into (2.12), it holds

$$\|S^{T/2}U\|_F = \sqrt{\sum_{i=1}^{N} \lambda_i}.$$

The relation between the "energy" of the matrix and the eigenvalues, which is corresponding to the proper orthogonal modes, can be seen as a hint of the choice of the optimal number L of the POD modes, since the goal is to gain most information of the data matrix by using the least number of POD modes. For this reason, one defines

$$E(l) = \sqrt{\frac{\sum_{i=1}^{L} \lambda_i}{\sum_{i=1}^{N} \lambda_i}} \tag{2.14}$$

as an indication of the magnitude of the preserved "information" by POD modes.
Moreover, for the relation between the errors of the snapshots and the POD modeling and the eigenvalues, the following proposition is introduced.

Proposition 2.2.1. *Consider*

$$\sum_{m=1}^{M} \left|\left| u\left(t_m, \mathbf{x}\right) - \sum_{l=1}^{L} \alpha_l\left(t_m\right) \varphi_l\left(\mathbf{x}\right) \right|\right|^2 . \tag{2.15}$$

The sum of the error between the snapshots and the POD modelling is

$$\sum_{m=1}^{M} \left|\left| u\left(t_m, \mathbf{x}\right) - \sum_{l=1}^{L} \alpha_l\left(t_m\right) \varphi_l\left(\mathbf{x}\right) \right|\right|^2 = \sum_{l=L+1}^{D} \lambda_i. \tag{2.16}$$

Proof. The basic idea of proof could be found in [22] for the canonical inner product in $\mathbb{R}^d$. Here, the proof for arbitrary inner product will be given.

$$\begin{aligned} \lambda_l &\overset{(2.2)}{=} \lambda_l(\varphi_l, \varphi_l) \overset{(2.7)}{=} \left(\sum_{m=1}^{M} (u_m, \varphi_l) u_m, \varphi_l \right) = \sum_{m=1}^{M} (u_m, \varphi_l)(u_m, \varphi_l) \\ &= \sum_{m=1}^{M} (u_m, \varphi_l)^2. \end{aligned} \tag{2.17}$$

Inserting (2.3), (2.4) into (2.15), and using the orthonormality of $\{\varphi_i\}_{l=1}^{D}$ yields

$$\begin{aligned} &\sum_{m=1}^{M} \left|\left| u\left(t_m, \mathbf{x}\right) - \sum_{l=1}^{L} \alpha_l\left(t_m\right) \varphi_l\left(\mathbf{x}\right) \right|\right|^2 \\ &= \sum_{m=1}^{M} \left|\left| \sum_{l=L+1}^{D} (u_m, \varphi_l) \varphi_l \right|\right|^2 \\ &= \sum_{m=1}^{M} \sum_{l=L+1}^{D} (u_m, \varphi_l)^2 \overset{interchange\, of\, finite\, sums}{=} \sum_{l=L+1}^{D} \sum_{m=1}^{M} (u_m, \varphi_l)^2 \\ &\overset{(2.17)}{=} \sum_{l=L+1}^{D} \lambda_l. \end{aligned}$$

□

As it is already mentioned, the matrices UU^TS and U^TSU share the same eigenvalues $\lambda_1, \ldots, \lambda_D$. For the sake of computational efficiency, it is generally considered to solve the eigenvalue problems with the matrix U^TSU in lieu of UU^TS. The approach of computing the eigenvalues of the matrix U^TSU and obtain the eigenvectors $\underline{\varphi_l}$ is called method of snapshots, it was first proposed in [20], see also [3].
By (2.4) and (2.8), it follows that

$$\alpha_l = U^TS\underline{\varphi_l} \in \mathbb{R}^M. \tag{2.18}$$

Multiplying (2.10) from left by the matrix U^TS, and inserting Eq.(2.18) into it, then the new eigenvalue problem is obtained

$$U^TSU\alpha_l = \lambda_l\alpha_l, \quad U^TSU \in \mathbb{R}^{M\times M}. \tag{2.19}$$

One can compute the eigenvectors $\{\alpha_l\}_{l=1}^L$ by solving the eigenvalue problem (2.19). It remains to induce the eigenvectors $\{\underline{\varphi_l}\}_{l=1}^L$ of the original eigenvalue problem (2.10) corresponding to $\{\alpha_l\}_{l=1}^L$.
Multiplying (2.19) from left by U leads to

$$UU^T SU\alpha_l = \lambda_l U\alpha_l.$$

It can be seen as a eigenvalue problem which has the same matrix $UU^T S$ as (2.10) but with different eigenvectors.
Considering the second constrained condition of Lagrangian function

$$(\varphi_l, \varphi_l) = 1$$

which implies

$$\|\varphi_l\| = 1, \quad l = 1, \ldots, L,$$

it is natural to set

$$\underline{\varphi_l} = \frac{1}{\|U\alpha_l\|} U\alpha_l$$

where $\|U\alpha_l\| = (\alpha_l^T U^T SU\alpha_l)^{1/2}$.
Multiplying (2.19) from left by α_l^T leads to

$$\alpha_l^T U^T SU\alpha_l = \lambda_l \alpha_l^T \alpha_l.$$

It follows that $\|U\alpha_l\| = \lambda_l^{1/2}(\alpha_l^T\alpha_l)^{1/2}$. Hence,

$$\underline{\varphi_l} = \frac{1}{\|U\alpha_l\|} U\alpha_l = \frac{1}{\sigma_l(\alpha_l^T\alpha_l)^{1/2}} U\alpha_l \tag{2.20}$$

with $\sigma_l = \sqrt{\lambda_l}$.
Inserting (2.20) and (2.19) into (2.10), one gets

$$UU^T S\underline{\varphi_l} = \frac{1}{\sigma_l(\alpha_l^T\alpha_l)^{1/2}} U(U^T SU\alpha_l) = \lambda_l \frac{1}{\sigma_l(\alpha_l^T\alpha_l)^{1/2}} U\alpha_l = \lambda_l \underline{\varphi_l}.$$

It shows that $\{\underline{\varphi_l}\}_{l=1}^L$ given in (2.20) indeed satisfy the original eigenvalue.

2.3 Applying the POD method on the fluctuations of functions

In practice, when applying the POD method to the equations with stationary Dirichlet boundary conditions

$$u(t, \mathbf{x}) = b(\mathbf{x}) \quad on[0, T] \times \partial\Omega, \quad for\ a\ certain\ value\ b(\mathbf{x}) \in \mathbb{R}^d,$$

often the fluctuations $\bar{u}(t, \mathbf{x})$ of the function $u(t, \mathbf{x})$ instead of the function $u(t, \mathbf{x})$ itself is used, for ease of observation. For this reason, firstly, one has to define the mean value of the snapshots $u_m = u_m(\mathbf{x}) = u(t_m, \mathbf{x})$.

$$\overline{u}(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} u_m(\mathbf{x}). \tag{2.21}$$

The fluctuations are obtained by subtracting the mean value $\bar{u}(\mathbf{x})$ from the snapshots u_m

$$\hat{u}_m = u_m - \overline{u}(\mathbf{x}), \quad m = 1, \ldots, M,$$

and the ensemble of the fluctuations of the snapshots is denoted as

$$\hat{V^h} = span\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_M\}.$$

By this means, a snapshot matrix with respect to the fluctuations $\bar{u}(t, \mathbf{x})$, which possess homogeneous Dirichlet boundary conditions, is obtained. And the formula for ROM from fluctuations can be represented as

$$u^L = \bar{u}(\mathbf{x}) + \sum_{l=1}^{L} \hat{\alpha}_l(t)\hat{\varphi}_l(\mathbf{x}).$$

Similarly to the case of $u(t, \mathbf{x})$, one gets the objective basis functions $\{\hat{\varphi}_l(\mathbf{x}) : \Omega \to \mathbb{R}^d\}_{l=1}^L$ by solving the optimization problem

$$\underset{\hat{\varphi}_1,\ldots,\hat{\varphi}_L}{\arg\min} \sum_{m=1}^{M} \left\| \hat{u}_m - \sum_{l=1}^{L} \hat{\alpha}_l(t_m)\hat{\varphi}_l(\mathbf{x}) \right\|^2.$$

From which comes again the eigenvalue problem

$$\hat{U}\hat{U}^T S\underline{\hat{\varphi}_l} = \lambda_l' \underline{\hat{\varphi}_l}, \quad l = 1, \ldots, L, \tag{2.22}$$

where $\hat{U} \in \mathbb{R}^{N\times M}$ is the snapshot matrix of the fluctuations with $(\hat{U})_{nm} = \hat{u}(t_m, \mathbf{x}_n)$. For saving the computations, in practice, one solves the eigenvalue problems

$$\hat{U}^T S\hat{U}\hat{\alpha}_l = \lambda_l' \hat{\alpha}_l, \quad \hat{U}^T S\hat{U} \in \mathbb{R}^{M\times M} \tag{2.23}$$

instead. Thereby, the eigenvectors $\{\underline{\hat{\varphi}_l}\}_{l=1}^L$ of the eigenvalue problem (2.23) are achieved with the relation

$$\underline{\hat{\varphi}_l} = \frac{\hat{U}\hat{\alpha}_l}{\|\hat{U}\hat{\alpha}_l\|}, \quad l = 1, \ldots, L, \tag{2.24}$$

and hence the wanted basis function is obtained

$$\hat{\varphi}_l = \sum_{n=1}^{N} \underline{\hat{\varphi}_l}(\mathbf{x}_n)\psi_n(\mathbf{x}),$$

where $\underline{\hat{\varphi}_l} = (\hat{\varphi}_l(\mathbf{x}_1), \hat{\varphi}_l(\mathbf{x}_2), \ldots, \hat{\varphi}_l(\mathbf{x}_N))^T$ denotes the vector of the coefficients of $\hat{\varphi}_l$ with respect to the nodal basis $\{\psi_n\}_{n=1}^N$.

3 Scalar convection-diffusion-reaction equations

In this thesis, we discuss the applications of the POD method introduced in Chapter 2 on scalar convection-dominated convection-diffusion-reaction equations.

3.1 The model problem

The convection-diffusion-reaction equation, as the name suggests, consists of the convection, diffusion, and reaction terms. It describes physical phenomena, which are due to the reaction, diffusion and convection processes of a certain chemical species, and models the concentration of these chemical species in the convection-diffusion-reaction process, see [6].

Definition 3.1.1. *General form of scalar convection-diffusion-reaction equations with homogeneous Dirichlet boundary condition.*

Let $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$, be a bounded domain with Lipschitz boundary $\partial\Omega$. A time-dependent convection-diffusion-reaction equation with homogeneous Dirichlet boundary condition is given by

$$\begin{aligned} \partial_t u - \varepsilon\Delta u + \mathbf{b}\cdot\nabla u + cu &= f \quad && in \quad (0,T]\times\Omega, \\ u &= 0 \quad && on \quad [0,T]\times\partial\Omega, \\ u(0,\mathbf{x}) &= u_0(\mathbf{x}) \quad && in \quad \Omega, \end{aligned} \tag{3.1}$$

where $\varepsilon > 0$ is a constant diffusion coefficient, $u_0(\mathbf{x})$ is a given initial condition. The functions $\mathbf{b}(t,\mathbf{x})$ and $c(t,\mathbf{x})$ denote the given convection and reaction field, respectively. It is assumed that $\mathbf{b},\ c \in L^2((0,T);\ L^\infty)$.

Remark 3.1.1. The convection-dominated problem. As its name implies, the convection-dominated problems arise in physical processes where convection essentially dominates diffusion, i.e., $\varepsilon \ll \|\mathbf{b}\|$. It appears in many situations, e.g., from nature or some technical applications. The solutions of the convection-dominated problem generally possess very small but important structures (or scales), such as the sharp layers, especially at boundaries. Most of these small structures of the solutions are not resolved by the standard numerical methods, since the grids of the domain in common use for these numerical methods are not fine enough to represent these small structures, hence special methods are needed. More discussions can be found in [9][21] ◁

3.2 The weak formulation

Remark 3.2.1. Motivation. In general, one cannot expect a classical solution of (3.1), since for the existence of a classical solution, the parameters have to be sufficiently smooth,

and in higher dimensions, the domain has to satisfy certain regularity conditions. Such smoothness and regularity conditions are not satiesfied in applications. In the weak formulation, such regularity assumptions can be reduced by integration and the transfer of derivatives to the test function. For problem (3.1), due to the Dirichlet boundary condition, the integral on the boundaries vanish, and one can search for the weak solutions in the ansatz space $H_0^1(\Omega)$ and the same for the test space. Besides, the finite element methods, which will be used for discretizting the partial differential equations, are based on a weak formulation. For more details, see [8][9]. ◁

Consider problem (3.1). To get the weak form, one multiplies the differential equation with an appropriate function $v(\mathbf{x})$, with $v = 0$ on $\partial\Omega$, integrating the resulting equation on Ω and using integration by parts in space yields

$$\begin{aligned}
&\int_\Omega (\partial_t u - \varepsilon\Delta u + \mathbf{b}\cdot\nabla u + cu)(\mathbf{x})v(\mathbf{x})d\mathbf{x}\\
&= \int_\Omega (\partial_t u \cdot v)(\mathbf{x})d\mathbf{x} - \int_{\partial\Omega} -\varepsilon(\nabla u\cdot\mathbf{n})v(\mathbf{s})d\mathbf{s} + \int_\Omega (\varepsilon\nabla u\cdot\nabla v + (\mathbf{b}\cdot\nabla u + cu)v)(\mathbf{x})d\mathbf{x}\\
&= \int_\Omega (\partial_t u\cdot v + \varepsilon\nabla u\cdot\nabla v + (\mathbf{b}\cdot\nabla u + cu)v)(\mathbf{x})d\mathbf{x}\\
&= \int_\Omega f(\mathbf{x})v(\mathbf{x})d\mathbf{x}.
\end{aligned}$$

Here, $\mathbf{n}$ is the outward pointing unit normal vector on $\partial\Omega$.

Definition 3.2.1. *Weak formulation of* (3.1). *Let* $\mathbf{b}, c \in L^\infty(\Omega)$, $f \in L^2(\Omega)$, *and* $V = H_0^1(\Omega)$. *Then, the weak form of* (3.1) *reads as follows: Find* $u \in V$ *such that*

$$(\partial_t u, v) + (\varepsilon\nabla u, \nabla v) + (\mathbf{b}\cdot\nabla u, v) + (cu, v) = (f, v) \qquad \forall v \in V \tag{3.2}$$

with the initial condition $u(0,\mathbf{x}) = u_0(\mathbf{x})$, *and* $(\cdot,\cdot)$ *denotes the inner product in* $L^2(\Omega)$. *A solution of* (3.2) *is called variational or weak solution. The space, in which the solution is searched, is called solution space. The functions* $v(\mathbf{x})$ *are called test functions, and the space* V *is the test space.*

To discuss the existence and uniqueness of the solution of the weak problem (3.2), firstly consider the continuous bilinear form $a(\cdot,\cdot) : V \times V \to R, \quad V = H_0^1(\Omega)$,

$$a(u,v) = \int_\Omega (\varepsilon\nabla u\cdot\nabla v + (\mathbf{b}\cdot\nabla u + cu)v)(\mathbf{x})d\mathbf{x}. \tag{3.3}$$

Proposition 3.2.1. *The coercivity of* (3.3). *Let* $\mathbf{b} \in C^1(\bar\Omega)$, $c \in C(\bar\Omega)$. *Assume there is a certain constant* c_0 *such that* $0 < c_0 \leqslant (-\frac{1}{2}\nabla\cdot\mathbf{b} + c)(t,\mathbf{x})$. *Then, the bilinear form* (3.3) *is coercive.*

Proof. Choose the test function $u = v \in V$, insert it into (3.3)

$$a(v,v) = \int_\Omega (\varepsilon\nabla v\cdot\nabla v + (\mathbf{b}\cdot\nabla v + cv)v)(\mathbf{x})d\mathbf{x}$$

Consider the term $\int_\Omega \mathbf{b}(\mathbf{x})\cdot\nabla v(\mathbf{x})v(\mathbf{x})d\mathbf{x}$, applying integration by parts with the homogeneous Dirichlet condition and using the product chain rule yields

$$\begin{aligned}
\int_\Omega \mathbf{b}(\mathbf{x})\cdot\nabla v(\mathbf{x})v(\mathbf{x})d\mathbf{x} &= -\int_\Omega \nabla\cdot(\mathbf{b}(\mathbf{x})v(\mathbf{x}))v(\mathbf{x})d\mathbf{x}\\
&= -\int_\Omega (\nabla\cdot\mathbf{b}(\mathbf{x}))v(\mathbf{x})v(\mathbf{x})d\mathbf{x} - \int_\Omega \mathbf{b}(\mathbf{x})\cdot\nabla v(\mathbf{x})v(\mathbf{x})d\mathbf{x},
\end{aligned}$$

which implies

$$\int_\Omega \mathbf{b}(\mathbf{x}) \cdot \nabla v(\mathbf{x}) v(\mathbf{x}) d\mathbf{x} = -\frac{1}{2} \int_\Omega (\nabla \cdot \mathbf{b}(\mathbf{x})) v(\mathbf{x}) v(\mathbf{x}) d(\mathbf{x}).$$

Hence

$$a(v,v) = \int_\Omega \left(\varepsilon (\nabla v)^2 + \left(-\frac{\nabla \cdot \mathbf{b}}{2} + c \right) v^2 \right) (\mathbf{x}) d\mathbf{x}$$

By the assumption that

$$0 < c_0 \leqslant (-\frac{1}{2} \nabla \cdot \mathbf{b} + c)(t, \mathbf{x}),$$

one obtains

$$a(v,v) \geqslant \varepsilon \|v\|^2_{L^2(\Omega)} = \varepsilon \|v\|^2_V, \quad \forall v \in V, \tag{3.4}$$

and this inequality is equivalent to the coercivity of $a(\cdot, \cdot)$.

□

Remark 3.2.2. Weaker assumptions of convection and reaction field $\mathbf{b}$ and c. The assumptions in Proposition 3.2.1 for $\mathbf{b}$ and c can be relaxed by $\mathbf{b}$, $\nabla \mathbf{b}$, $c \in Ł^\infty(\Omega)$, since under these assumptions all the integrals are still well defined. ◁

For more details about the properties of $a(\cdot, \cdot)$, it is referred to [9].

Corollary 3.2.1. *Existence and uniqueness of a solution of the weak problem* (3.2). *Let the assumptions of Proposition.3.2.1 be satisfied. Then,* (3.2) *has a unique solution.*

Proof. The proof of this corollary is similar to the proof of existence and uniqueness of the solution for the time-dependent Navier-Stokes equations in [10], and for the general linear second-order parabolic equations in [18].
Firstly consider the weak form of the convection-diffusion-reaction equations (3.2) in finite-dimensional space. Since $V = H^1_0(\Omega)$, it admits an orthonormal basis $\{\Phi_i\}_{i \geqslant 1}$. Define the finite-dimensional subspace

$$V^N = span\{\Phi_j\}_{j=1}^N \subset V$$

and apply the Galerkin method to (3.2) in V^N yields the approximate problems

$$(\partial_t u^N, v^N) + (\varepsilon \nabla u^N, \nabla v^N) + (\mathbf{b} \cdot \nabla u^N, v^N) + (c u^N, v^N) = (f, v^N) \qquad \forall v^N \in V^N, \tag{3.5}$$

and the initial condition $u^N(0)$ is the orthogonal projection in $L^2(\Omega)$ of u_0 on V^N. It is clear that (3.5) holds in particular for each basis functions Φ_j.
By the property of orthonormal basis, u^N can be represented as

$$u^N(t) = \sum_{j=1}^N \alpha_j^N(t) \Phi_j \tag{3.6}$$

Inserting (3.6) into (3.5), one obtains the system of ordinary differential equations

$$\begin{cases} M \frac{d}{dt} \alpha^N + a \alpha^N = \mathbf{F}, \\ M \alpha^N(0) = \alpha_{\mathbf{0}}, \end{cases} \tag{3.7}$$

with

$$M := (\Phi_i, \Phi_j), \quad a_{ij} := (\varepsilon \nabla \Phi_i, \nabla \Phi_j) + (\mathbf{b} \cdot \nabla \Phi_i + c \Phi_i, \Phi_j),$$

$$\alpha^N = \{\alpha_j^N\}_{j=1}^N, \quad F_j := (f, \Phi_j), \quad \alpha_{0,j} := (u_0, \Phi_j), \quad i, j = 1, \ldots, N.$$

By the *Carathéodory* theorem, there is a local solution $\alpha^N \in H^1(0,t)$ with $0 \leqslant t_n \leqslant T$, i.e., $u^N \in H^1(0,t;V)$.
Secondly, the global existence is shown by proving the boundedness. Choosing $u^N(t)$ in (3.5) as test function for arbitrary $t \in (0,T)$ one obtains

$$\left(\frac{d}{dt}u^N(t), u^N(t)\right) + a(u^N(t), u^N(t)) = (f, u^N(t)). \tag{3.8}$$

Using the product chain rule for first term, inserting (3.4) and Young's inequality yields

$$\frac{1}{2}\frac{d}{dt}\|u^N(t)\|^2_{L^2(\Omega)} + \varepsilon\|u^N(t)\|^2_V \leqslant \frac{1}{2\varepsilon}\|f\|^2_{L^2(\Omega)} + \frac{\varepsilon}{2}\|u^N(t)\|^2_{L^2(\Omega)},$$

which gives

$$\frac{d}{dt}\|u^N(t)\|^2_{L^2(\Omega)} + \varepsilon\|u^N(t)\|^2_V \leqslant \frac{1}{\varepsilon}\|f\|^2_{L^2(\Omega)}. \tag{3.9}$$

Integrating (3.9) in $[0,t]$, $t \in (0,T]$ yields

$$\|u^N(t)\|^2_{L^2(\Omega)} + \varepsilon\int_0^t \|u^N(t)\|^2_V d\tau \leqslant \int_0^t \|u_0\|^2_{L^2(\Omega)} d\tau + \frac{1}{\varepsilon}\|f\|^2_{L^2(\Omega)} d\tau. \tag{3.10}$$

Thus u^N is bounded in $L^\infty(0,T;L^2(\Omega)) \cap L^2(0,T;V)$, and the local existence of u^N can be extended to $t = T$.
Next step is to choose a subsequence $\{u^{N_l}\}_{N_l \geqslant 1}$ of $\{u^N\}_{N \geqslant 1}$ such that

$$\begin{aligned} u^{N_l} &\overset{*}{\rightharpoonup} u \quad in \quad L^\infty(0,T;L^2(\Omega)), \\ u^{N_l} &\rightharpoonup u \quad in \quad L^2(0,T;V). \end{aligned}$$

as $N_l \to \infty$. Here, the notations $\overset{*}{\rightharpoonup}$ and $\rightharpoonup$ mean weakly* convergent and weakly convergent, respectively (see, e.g., [10], pp. 270-271).
This implies that there exits $u \in L^\infty(0,T;L^2(\Omega)) \cap L^2(0,T;V)$ such that

$$\int_0^T \left(u^{N_l}(t), \phi(t)\right) \rightharpoonup \int_0^T (u(t), \phi(t)) \quad \forall \phi \in L^\infty(0,T;L^2(\Omega)),$$

$$\int_0^T \left(\nabla u^{N_l}(t), \psi(t)\right) \rightharpoonup \int_0^T (\nabla u(t), \psi(t)) \quad \forall \psi \in L^2(0,T;L^2(\Omega))$$

as $N_l \to \infty$.
Let $\Psi(t) \in C_0^1((0,T))$ and arbitrary $v \in V$, for the first term of (3.5), multiplying this term by $\Psi(t)$, apply integration by parts and with the property of weak convergence

$$\begin{aligned} \lim_{N_l \to \infty} \int_0^T \left(\partial_t u^{N_l}(t), v\right) \Psi(t) dt &= -\lim_{N_l \to \infty} \int_0^T \left(u^{N_l}(t), v\right) \frac{d}{dt}\Psi(t) dt \\ &= -\int_0^T (u(t), v) \frac{d}{dt}\Psi(t) dt \\ &= \int_0^T (\partial_t u(t), v) \Psi(t) dt. \end{aligned} \tag{3.11}$$

For the rest terms at the left hand side of (3.5), let $\Psi(t) \in C_0^1([0,T])$ and arbitrary $v \in V$, it holds

$$\begin{aligned}\lim_{N_l\to\infty}\int_0^T a(u^{N_l},v)\Psi(t) &= \lim_{N_l\to\infty}\int_0^T \left((\varepsilon\nabla u^{N_l},\nabla v)+(\mathbf{b}\cdot\nabla u^{N_l}+cu^{N_l})\right)\Psi(t)dt \\ &= \int_0^T \left((\varepsilon\nabla u,\nabla v)+(\mathbf{b}\cdot\nabla u+cu)\right)\Psi(t)dt \\ &= \int_0^T a(u,v).\end{aligned} \tag{3.12}$$

Next step is to show the satisfaction of the initial condition. The proof is quite similar to the proof of Lemma 6.18 in [10]. Choose $\Psi(t) = \frac{T-t}{T}$, and arbitrary $v^{N_l} \in V^N$. By the fundamental theorem of calculus and using the product chain rule yields

$$\begin{aligned}-(u(0),v) &= -\lim_{N_l\to\infty}\left(u(0),v^{N_l}\right) \\ &= \lim_{N_l\to\infty}\left(u(t),v^{N_l}\right)\frac{T-t}{T}\Big|_{t=0}^{T} \\ &= \lim_{N_l\to\infty}\int_0^T \frac{d}{dt}\left((u(t),v^{N_l})\frac{T-t}{T}\right)dt \\ &= \lim_{N_l\to\infty}\left(\int_0^T(\partial_t u(t),v^{N_l})\frac{T-t}{T}dt-\frac{1}{T}\int_0^T\left(u(t),v^{N_l}\right)dt\right).\end{aligned} \tag{3.13}$$

From Eq.(3.2), one obtains that

$$\left(\partial_t u(t),v^{N_l}\right) = \left(f,v^{N_l}\right) - a(u(t),v^{N_l}). \tag{3.14}$$

As (3.5) is satisfied for each Φ_j, it is also satisfied for v^{N_l}

$$(f,v^{N_l}) = (\partial_t u^{N_l},v^{N_l}) + a(u^{N_l},v^{N_l}). \tag{3.15}$$

Inserting (3.14) and (3.15) into (3.13) yields

$$\begin{aligned}&-\lim_{N_l\to\infty}\left(u(0),v^{N_l}\right) = \lim_{N_l\to\infty}\left(\int_0^T\left((f,v^{N_l})-a(u,v^{N_l})\right)\frac{T-t}{T}dt-\frac{1}{T}\int_0^T(u(t),v^{N_l})dt\right) \\ &= \lim_{N_l\to\infty}\int_0^T(\partial_t u^{N_l},v^{N_l})\frac{T-t}{T}dt+\lim_{N_l\to\infty}\int_0^T a(u^{N_l}-u,v^{N_l})\frac{T-t}{T}dt \\ &-\frac{1}{T}\lim_{N_l\to\infty}\int_0^T(u(t),v^{N_l})dt.\end{aligned} \tag{3.16}$$

Consider now $N_l \to \infty$. By (3.12) it follows that the term

$$\lim_{N_l\to\infty}\int_0^T a(u^{N_l}-u,v^{N_l})\frac{T-t}{T}dt$$

vanishes. Applying integration by parts in time and (3.11) to the rest terms of the right hand side of Equation (3.16) yields

$$\begin{aligned}-(u(0),v) &= -\lim_{N_l\to\infty}(u(0),v^{N_l}) \\ &= \lim_{N_l\to\infty}(u^{N_l},v^{N_l})\frac{T-t}{T}\Big|_{t=0}^{T}+\frac{1}{T}\lim_{N_l\to\infty}\int_0^T\left((u^{N_l},v^{N_l})-(u,v^{N_l})\right)dt \\ &= -\lim_{N_l\to\infty}(u^{N_l}(0),v^{N_l}).\end{aligned}$$

Hence the limit function $u(t)$ satisfies the initial condition.
It remains to check the regularity of the time derivative term $\partial_t u(t)$. Define

$$Au \in V', \quad (Au, v) = a(u, v),$$

from (3.2) one obtains

$$(\partial_t u(t), v) + (Au, v) = (f, v), \quad \forall v \in V.$$

Since $Au \in L^2(0, T; V')$ when $u \in L^2(0, T; V)$ (see, e.g., [6], pp 156-159), and by hypothesis $f \in L^2(0, T; V')$, $\partial_t u \in L^2(0, T; V')$.
Applying the same approach of the derivation of (3.10), the following inequality holds for $u(t)$

$$\|u(t)\|^2_{L^2(\Omega)} + \varepsilon \int_0^t \|u(t)\|^2_V d\tau \leqslant \int_0^t \|u_0\|^2_{L^2(\Omega)} d\tau + \frac{1}{\varepsilon} \int_0^T \|f\|^2_{L^2(\Omega)} d\tau \tag{3.17}$$

and the uniqueness results from (3.17), since if there exits another solution v, assume $w = u - v$, it is clear that w also satisfies (3.2) with $w(0) = 0$ and $f = 0$ in the sense of distribution. Inserting w into (3.17) yields

$$\|w(t)\|^2_{L^2(\Omega)} + \varepsilon \int_0^t \|w(t)\|^2_V d\tau \leqslant 0, \tag{3.18}$$

This implies $\|w\|_{L^\infty(0,T;L^2(\Omega))} = 0$ and $\|w\|_{L^2(0,T;V)} = 0$. Hence there exits unique solution $u(t) \in L^2(0, T; V) \cap L^\infty(0, T; L^2(\Omega))$. □

3.3 The Galerkin method

Remark 3.3.1. The discretization of (3.2). Since the weak formulation (3.2) is time-dependent, the discretizations are needed both for time and space. In this thesis, the method of Rothe, which applies first the discretization in time and then in space, is considered. ◁

Let $t_1, \ldots, t_n$ represents the discrete times in $[0, T]$, where $t_0 = 0$ and $t_n = T$. Let u_k be the solution at time t_k, $k = 1, 2, \ldots, n$. For simplifying, the equidistant time step $\Delta t = t_k - t_{k-1}$ will be assumed.
Since the diffusive term may cause a stiff problem, the explicit Euler method becomes very insufficient to achieve stable solutions unless very small time steps Δt are applied, which is impractical. For the reasons of easy implementation and the good stability of the solutions, the one-step backward Euler scheme is considered to discretize the problem (3.2) in time. It is given as follows: For k=1,2,..., find $u_k \in V$ such that $\forall v \in V$

$$(u_k, v) + \Delta t[(\varepsilon \nabla u_k, \nabla v) + (\mathbf{b}_k \cdot \nabla u_k + c u_k, v)] = (u_{k-1}, v) + \Delta t(f_k, v). \tag{3.19}$$

Remark 3.3.2. The Galerkin finite element formulation. Let $\{\mathcal{T}^h\}$ be a family of regular triangulations consisting of mesh cells $\{K\}$. For simplifying, the triangulations are assumed to be quasi-uniform, and h is the diameter of all mesh cells K. Let $V^h \subset V$ be conforming finite element space. Then the Galerkin finite element formulation of problem (3.19), which replaces the space V in (3.19) by V^h, reads as follows: Find $u^h_k \in V^h : [0, T] \to V^h, \quad h = 1, 2, \ldots$ such that

$$(u^h_k - u^h_{k-1}, v^h) + \Delta t[(\varepsilon \nabla u^h_k, \nabla v^h) + (\mathbf{b}_k \cdot \nabla u^h_k + c u^h_k, v^h)] = \Delta t(f_k, v^h) \tag{3.20}$$

for all $v^h \in V^h$ and $u^h(0, \mathbf{x}) = u^h_0(\mathbf{x}) \in V^h$, where $u^h(0, \mathbf{x})$ is an approximation of $u(0, \mathbf{x})$, for example, an appropriate interpolation or a projection. ◁

3.4 The Streamline-Upwind Petrov-Galerkin (SUPG) method

It is well known that for small ε, the Galerkin method fails to compute useful numerical solutions, since the computed solutions are spoiled globally with spurious oscillations. As it is mentioned in Remark 3.2.1, the solution usually possesses structures like layers, particularly at boundaries. Those structures, which are not resolved by the Galerkin finite element space, are important. For more discussions, see e.g., [9]. For this reason, one considers the Streamline-Upwind Petrov-Galerkin (SUPG) method.

Remark 3.4.1. Introduction of the Streamline-Upwind Petrov-Galerkin (SUPG) method. The Streamline-Upwind Petrov-Galerkin (SUPG) method is a very popular stabilized finite element method, which is proposed in [2]. It combines the ideas of using the Galerkin discretization and the minimization of the residual. ◁

Definition 3.4.1. *Streamline-Upwind Petrov-Galerkin (SUPG) method.*
The Streamline-Upwind Petrov-Galerkin (SUPG) finite element method has the form: Find $u^h \in V^h$*, such that*

$$a^h(u^h, v^h) = f^h(v^h) \quad \forall v^h \in V^h \tag{3.21}$$

with

$$a^h(v,w) := a(v,w) + \sum_{K\in\mathcal{T}} \int_K \delta_K \left(-\varepsilon\Delta v + \mathbf{b}\cdot\nabla v + cv\right)(\mathbf{b}\cdot\nabla w)(\mathbf{x})d\mathbf{x},$$
$$f^h(w) := (f,w) + \sum_{K\in\mathcal{T}} \int_K \delta_K f(\mathbf{b}\cdot\nabla w)(\mathbf{x})d\mathbf{x}.$$

Here, $\{\delta_K\}$ *are user-chosen weights, which are called stabilization parameters or SUPG parameters.*

Definition 3.4.2. *SUPG (SD) norm. Let*

$$-\frac{1}{2}\nabla\cdot\mathbf{b}(\mathbf{x}) + c(\mathbf{x}) \geqslant c_0 > 0.$$

be satisfied. For $\forall v \in V^h$*, the SUPG inner product is defined by*

$$(u,v)_{SD} = \varepsilon(\nabla u, \nabla v) + c(u,v) + \sum_{K\in\mathcal{T}^h} (\delta_K^{1/2}\mathbf{b}\cdot\nabla u, \delta_K^{1/2}\mathbf{b}\cdot\nabla v) \tag{3.22}$$

and the corresponding SUPG norm by

$$|||v|||_{SUPG} := \left(\varepsilon|v|_1^2 + c_0\|v\|_0^2 + \sum_{K\in\mathcal{T}^h} \|\delta_k^{1/2}(\mathbf{b}\cdot\nabla v)\|_{0,K}^2\right)^{1/2}, \tag{3.23}$$

where $\|\cdot\|_{0,K}$ *denotes the norm in* $L^2(K)$*.*

Remark 3.4.2. Concerning the SUPG method.

- The SUPG method can be considered as as a Petrov-Galerkin method with the test space
$$span\{w(\mathbf{x}) + \sum_{k\in\mathcal{T}^h} \delta_K \mathbf{b}(\mathbf{x})\cdot\nabla w(\mathbf{x})\}.$$

- The SUPG method introduces artificial diffusion only in streamline direction $\mathbf{b}(\mathbf{x}) \cdot \nabla w(\mathbf{x})$. From this property, the SUPG method is also called "Streamline Diffusion FEM (SDFEM)" method.

- δ_K is called stabilization parameters or SUPG parameters. Normally, it is a general function, but in practice it is often chosen as a piecewise constant function.

- For more details, see [19].

◁

Remark 3.4.3. Choice of the stabilization parameter δ_K. It is well known that for steady-state problems, the stabilization parameter is chosen to depend on the mesh width such that $\delta_K = \mathcal{O}(h)$ from finite element analysis, see [19]. In the case of time-dependent problem, an optimal error estimate was proved for $\delta_K = \mathcal{O}(\triangle t)$. However, for the special case that the time-continuous limit is approached, the SUPG stabilization parameter vanishes and one turns to Galerkin finite element method. This behavior is shown to be not correct by numerical studies, see [12]. For this reason, the simplified problems that the convection and diffusion being time independent was considered, and it can be proven that $\delta_K = \mathcal{O}(h)$ is the choice from optimal error estimates for certain temporary discretization schemes in [12]. Besides, as discussed in Remark 3.1.1, the difficulty in solving the convection-dominated problem arises from the application of the coarse mesh which is generally much coarser than the width of the sharp layer, rather than the time step. Hence it is more appropriate to choose the stabilization parameter depending on mesh width from the practical point of view, see [5]. For general time-continuous case, the choice for the stabilization parameter is an open problem. In summary, it is strongly suggested to define the stabilization parameters by $\delta_K = \mathcal{O}(h)$ for the SUPG method to compute the snapshots. See details in [12].
◁

Remark 3.4.4. The backward Euler method together with the SUPG method applying on (3.20) has the form:

$$\begin{aligned}(u_k^h - u_{k-1}^h, v^h) + \Delta t a_{SUPG,h}(u_k^h, v^h) =& \Delta t(f_k, v^h) + \Delta t \sum_{K\in\mathcal{T}^h} \delta_K (f_k, \mathbf{b}_k \cdot \nabla v^h)_K \\ &- \sum_{K\in\mathcal{T}^h} \delta_K (u_k^h - u_{k-1}^h, \mathbf{b}_k \cdot \nabla v^h)_K \end{aligned} \tag{3.24}$$

with the bilinear form

$$\begin{aligned} a_{SUPG,h}(u_k^h, v^h) =& (\varepsilon \nabla u_k^h, \nabla v^h) + (\mathbf{b}_k \cdot \nabla u_k^h + c u_k^h, v^h) \\ &+ \sum_{K\in\mathcal{T}^h} \delta_K (-\varepsilon \Delta u_k^h + \mathbf{b}_k \cdot u_k^h + c u_k^h, \mathbf{b}_k \cdot \nabla v^h)_K \end{aligned} \tag{3.25}$$

for all $v_h \in V_h$, and the initial condition $u_0^h(\mathbf{x}) = u^h(0, \mathbf{x})$ is given. Here, $\{\delta_k\}$ have to be chosen appropriately. Note that choosing one parameter $\delta_K = h$ for all mesh cells K is feasible only if the uniformity of the triangulations is assumed. ◁

3.5 ROMs based on the POD method.

In this section, to obtain the SUPG reduced order model, the POD method for the reduced order modeling of time-dependent problem is applied to (3.24). The SUPG-ROM was

proposed, see, e.g., [1]. Using the same notations as in Chapter 2, the POD basis functions $\{\hat{\varphi}_l\}_{l=1}^{L}$ are computed from the fluctuations $\hat{u}_m(\mathbf{x})$ instead of functions $u_m(t,\mathbf{x})$.
As it is defined in Chapter 2, $\{\hat{u}_1,\ldots,\hat{u}_M\}$ denote the snapshots of the fluctuations, which are obtained by solving Eq.(3.25), and

$$\hat{V}^h = span\{\hat{u}_1,\ldots,\hat{u}_M\}.$$

It is assumed that at least one of $\{\hat{u}_i\}_{i=1}^{M}$ is nonzero.
It is already introduced the way to calculate the POD modes in Chapter 2. Let V_L be the L-dimensional space spanned by the POD basis functions $\hat{\varphi}_1,\ldots,\hat{\varphi}_L$, for any $L \in \{1,\ldots,D'\}$, where D' is the dimension of $\hat{V}^h$. It is obvious that the relation $V_L \supseteq \hat{V}^h$ holds. Let $\hat{u}^L(t,\mathbf{x}) + \bar{u}(\mathbf{x})$ express the ROM approximation of the SUPG solution, where

$$\bar{u}(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} u_m(\mathbf{x})$$

and denote the ROM approximation at time instance t_k as $\hat{u}_k^l(\mathbf{x}) + \bar{u}(\mathbf{x})$, which will be written as $\hat{u}_k^L + \bar{u}$ for short.

Remark 3.5.1. The relation between the dimension of the snapshot space D and the dimension of the fluctuation space D'. Assume $\{\underline{b_1},\ldots,\underline{b_D}\}$ is a set of basis vectors of the snapshot space V^h, then $D' = D$ if $\bar{u}(\mathbf{x})$ and $\{\underline{b_i}\}$ are linear dependent, or else $D' = D+1$. ◁

To build the ROM based on the POD method, firstly express the reduced order solution u_k^L by the linear combination of the first L POD modes

$$u_k^L = \bar{u}(\mathbf{x}) + \sum_{l=1}^{L} \hat{\alpha}_l(t_k)\hat{\varphi}_l, \tag{3.26}$$

where $\{\hat{\alpha}_l(t_k)\}$ are the undetermined coefficients with respect to time.
Inserting (3.26) into (3.24), yields the backward/SUPG reduced order model: For $k = 1,2,\ldots$ find $\hat{u}_k^L = \sum_{l=1}^{L}\hat{\alpha}_l(t_k)\hat{\varphi}_l \in V_L$ such that $\forall v^L \in V_L$

$$\begin{aligned}
&\left(\sum_{l=1}^{L}\hat{\alpha}_l(t_k)\hat{\varphi}_l - \sum_{l=1}^{L}\hat{\alpha}_l(t_{k-1})\hat{\varphi}_l, v^L\right) + \Delta t a_{SUPG,h}\left(\sum_{l=1}^{L}\hat{\alpha}_l(t_k)\hat{\varphi}_l, v^L\right)\\
=&\Delta t(f_k, v^L) + \Delta t\sum_{K\in\mathcal{T}^l}\delta_K^r(f_k, \mathbf{b}_k\cdot\nabla v^L)_K\\
&-\sum_{K\in\mathcal{T}^l}\delta_K^r\left(\sum_{l=1}^{L}\hat{\alpha}_l(t_k)\hat{\varphi}_l - \sum_{l=1}^{L}\hat{\alpha}_l(t_{k-1})\hat{\varphi}_l, \mathbf{b}_k\cdot\nabla v^L\right)_K - \Delta t a_{SUPG,l}(\bar{u}, v^L).
\end{aligned} \tag{3.27}$$

Here, δ_K^r is the stabilization parameters from SUPG method in ROM simulations. For the choice of stabilization parameters from SUPG method in ROM simulations (noted as δ_k^r), the same form as the finite element SUPG discretization is preferable, i.e., $\delta_K^r = \mathcal{O}(h)$. The discussions can be found in [5].
By solving (3.27), the coefficients $\hat{\alpha}_l(t_k)$ are determined, and hence follows the reduced order solution u_k^L.

4 Numerical studies

For the following numerical examples which will be presented, the analytical solutions are already known. The code MooNMD [11] was used to run the numerical experiments.
Since in some examples in this section, the main topic is to compare the results, e.g., the POD modes and the ROMs, produced by different inner products, hence in these examples the different notations have to be considered. For convenience, the general notations without any specified explanations will be used for the case of L^2 inner product. One can evaluate the results by not only ROM solutions, but also the errors to some extent, e.g., the errors between the analytical solutions and the finite element solutions, and the errors between the analytical solutions and the ROM solutions. The form of the discrete analog of the errors is given, e.g., for the ROM solutions by

$$\frac{1}{M}\sum_{k=1}^{M}\|u_k - (\bar{u} + u_k^L)\|_* \tag{4.1}$$

where $\|\cdot\|_*$ denotes a certain norm, and u_k denotes the analytical solution of the continuous problem at time instance t_k, see [5].

4.1 Hump changing its height in two dimensions

This example is taken from [14]. It is defined in the domain $\Omega = (0,1)^2$ and (0,T)=(0,2). The coefficients in (3.1) are chosen to be $\varepsilon = 10^{-6}$, $\mathbf{b} = (2,3)^T$ and c=1.
This problem has the analytic solution of the form

$$\begin{aligned} u(t,x,y) =& \sin(\pi t)x(1-x)y(1-y) \\ &\times \left[\frac{1}{2} + \frac{arctan(2\varepsilon^{-1/2}(0.25^2 - (x-0.5)^2) - (y-0.5)^2))}{\pi}\right]. \end{aligned}$$

In this example, the comparisons of applying G-FEM and SUPG-FEM on this convection-dominated problem will be made, and the impact of using different spatial levels on both finite element solutions and the ROMs will be shown.Furthermore, applying different inner products on the computations of the POD modes and the ROMs will also be studied.
The backward Euler scheme was applied for time discretization with the length of time step Δt equal to 10^{-3}. Since the value of ε is very small, this problem is a convection-dominated problem. It is generally known that for small ε, as discussed in Remark 3.1.1, the solution of this problem has a layer by construction. Hence a useful stabilization method is needed.
Fig. 4.1 offers a comparison of using Galerkin finite element method (left) and the SUPG method (right).For simplification, the applications of these two methods for obtaining the snapshots will be denoted as G-FEM and SUPG-FEM. By observation of Fig. 4.1, it can be obviously seen that the solution of G-FEM is globally polluted and possesses a wide range of spurious oscillations, while the solution of SUPG-FEM is relatively smooth with

a few oscillations on the right upper side of the plane.

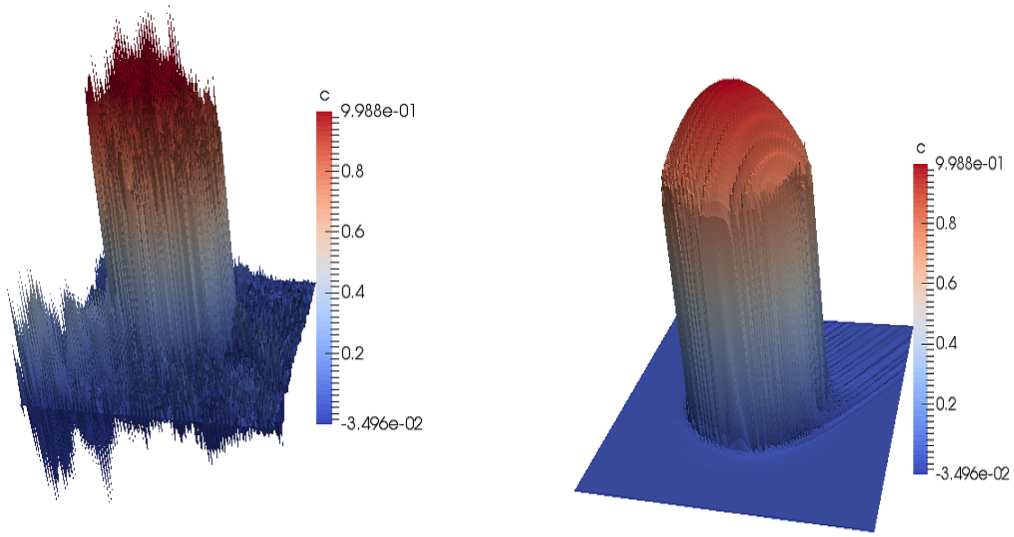


Figure 4.1: Example 4.1: Solutions of G-FEM and SUPG-FEM at t=0.5 with spatial level 7.

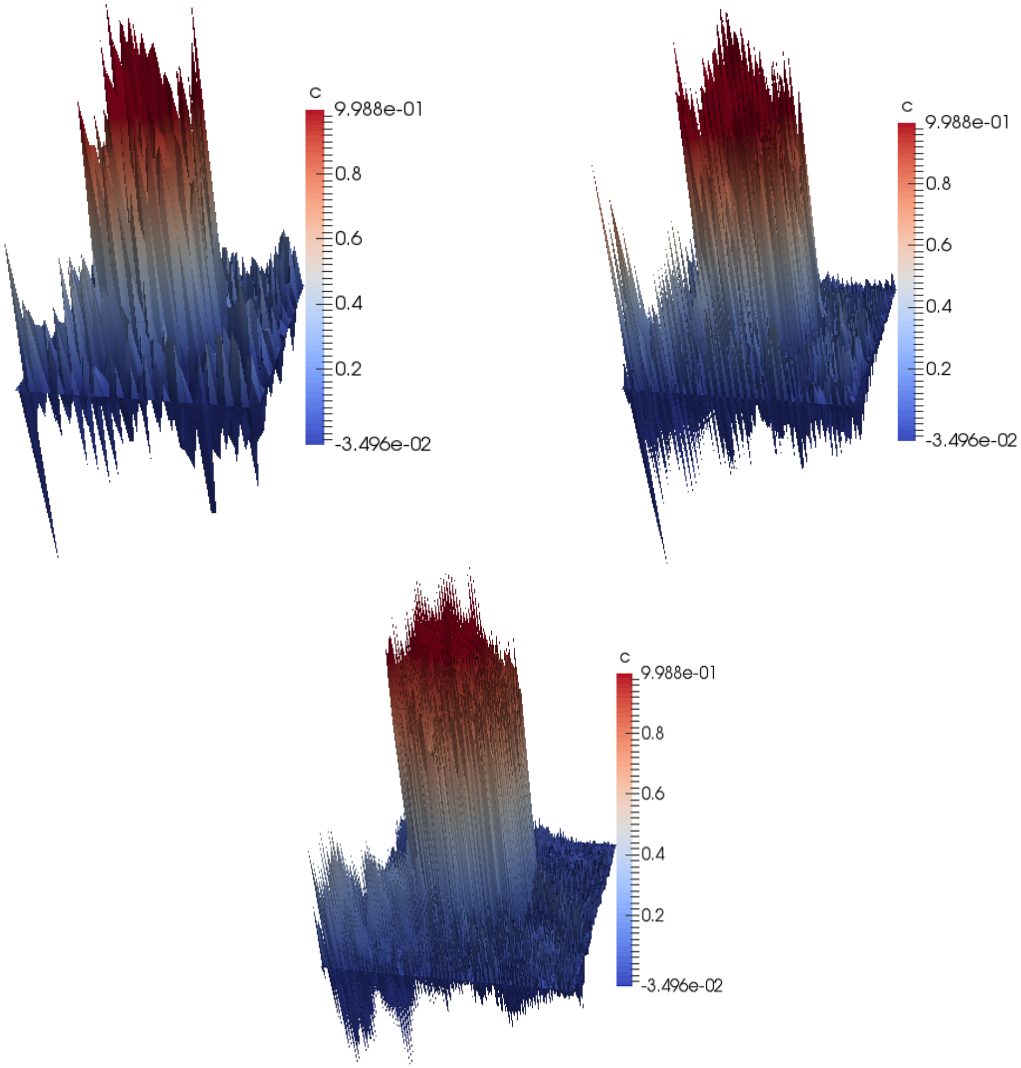


Figure 4.2: Example 4.1: Solutions of g5, g6 and g7 at t=0.5.

Fig. 4.2 and Fig. 4.3 present the results of G-FEM with spatial levels 5, 6 and 7 (denoted as g5, g6 and g7)(upper) and SUPG-FEM with spatial levels 5, 6 and 7 (denoted as s5, s6 and s7)(lower). From Fig. 4.3, it can be observed that, for SUPG-FEM, more details in the simulations are shown if higher spatial levels are applied, and the results become more accurate. The situation for G-FEM is different, there are always global spurious oscillations. It is worth mentioning that, when comparing the solutions of s5 and s6 in Fig. 4.3, though the spurious oscillations become smaller in general, the big oscillations appear at the boundary of the right upper side.

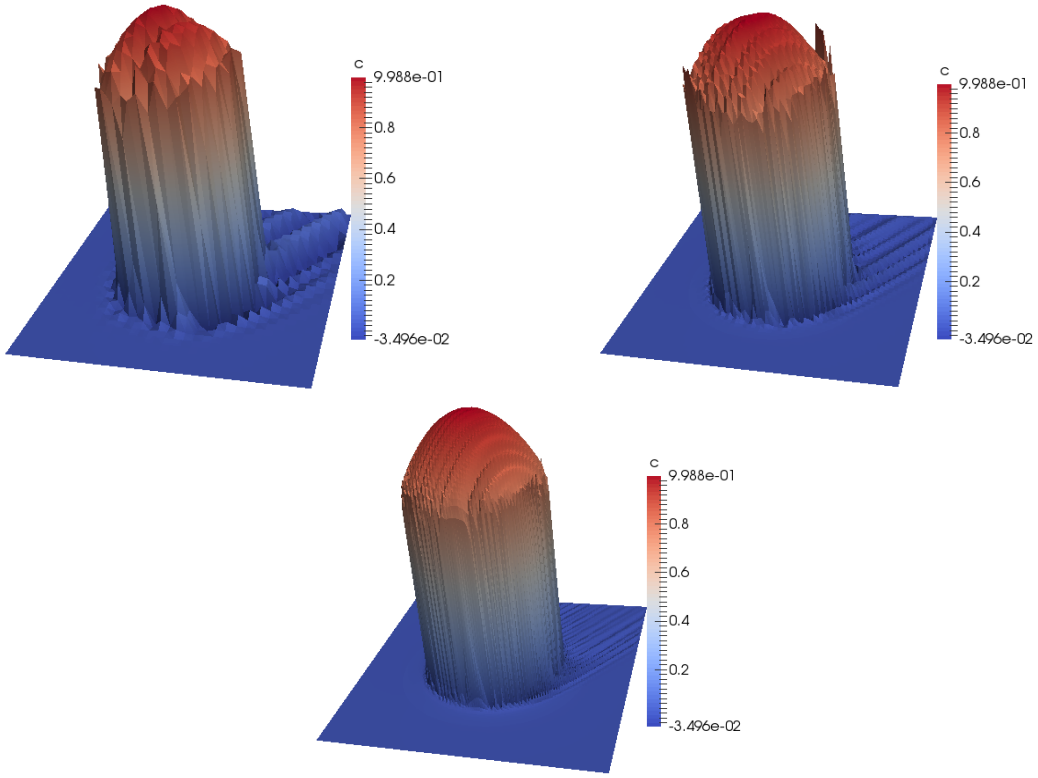


Figure 4.3: Example 4.1: Solutions of s5, s6 and s7 at t=0.5.

Since SUPG-FEM gives a better performance than the G-FEM in the simulations of convection-dominated problem, one computes the snapshots with SUPG-FEM, and takes every tenth snapshots such that there were 201 snapshots in total.
The application of the SUPG method on the reduced order problems will be denoted as SUPG-ROM for the purpose of distinguishing from SUPG-FEM. Similarly for G-ROM. Fig. 4.4 offers the temporal evolution of $L^2(\Omega)$ and $H^1(\Omega)$ errors for the solutions of SUPG-FEM with levels 4, 5, 6, 7 and the G-FEM with level 7. Observe the discrete $L^2(\Omega)$ errors in the left panel, it can be readily seen that the higher the level of grid for SUPG-FEM used, the smaller errors are produced. Furthermore, in spite of the fact that the finest level for G-FEM is used, compare to the solutions of s5, s6 and s7, the solution of g7 still causes the largest errors. The plot of the discrete $H^1(\Omega)$ errors in the right panel shows similar results, and the differences of errors for G-FEM and SUPG-FEM are even larger.

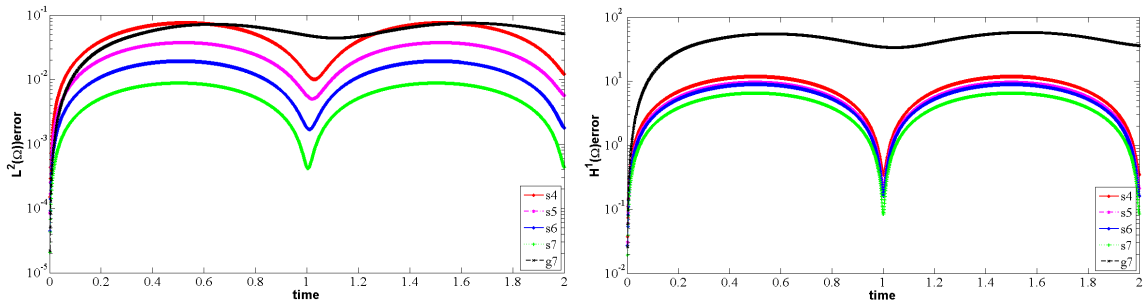


Figure 4.4: Example 4.1: $L^2(\Omega)$ (left) and $H^1(\Omega)$ (right) errors for s4, s5, s6, s7 and g7.

From the fact that there always exist spurious oscillations even if SUPG-FEM is applied, one cannot expect to obtain only one POD mode for the POD basis, since this implies the fact that this problem can be solved accurately by finite element method, and such a method is not known so far, see [13]. As a result, in this example, one obtains 4 POD modes for L^2 and H^1 inner products, and 14 POD modes for SD inner product. Fig. 4.5 shows the eigenvalues with respect to different inner product L^2, H^1, and SD with three different spatial levels. Among these POD modes, the eigenvalues corresponding to φ_1 for each inner product and spatial level dominate other eigenvalues notably. From (2.14), it is known that the first POD modes have taken nearly all "information" from the origin data, and the rest POD modes mainly come from the oscillations.

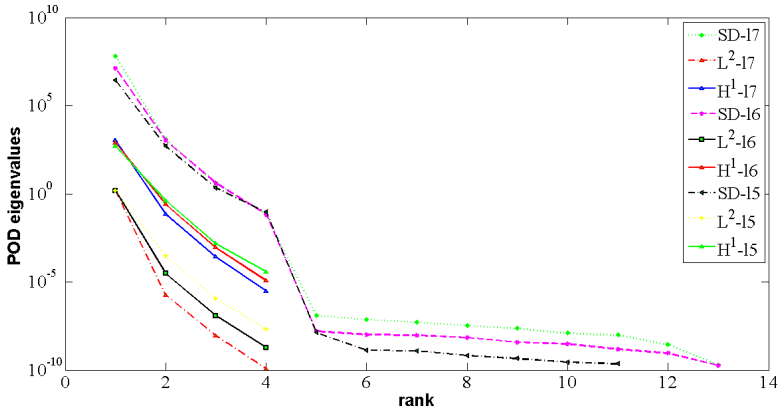


Figure 4.5: Example 4.1: POD eigenvalues for H^1, L^2, and SD inner product with spatial levels 5, 6, and 7.

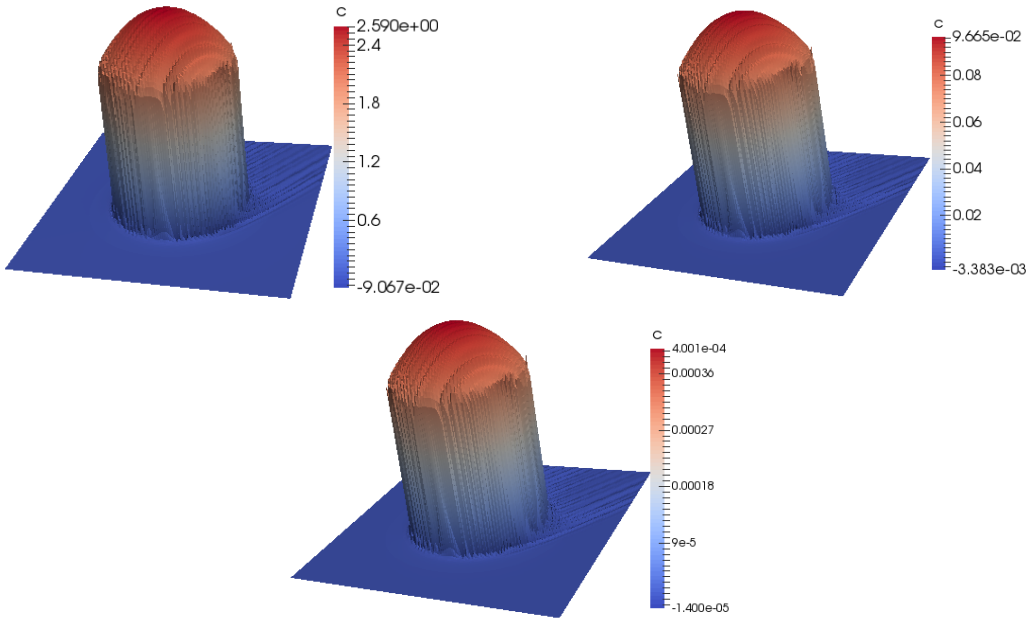


Figure 4.6: Example 4.1: $\mathcal{B}^1_{L^2}$(top left), $\mathcal{B}^1_{H^1}$(top right), $\mathcal{B}^1_{SD}$(bottom) for spatial level 7.

For simplification, the i-th POD modes computed from different inner products will be

written as $\mathcal{B}_m^i$, and the ROMs for different inner products will be denoted as ROM_m, $m \in \{L^2, H^1, SD\}$. Fig. 4.6 presents the pictures of $\mathcal{B}_{L^2}^1$, $\mathcal{B}_{H^1}^1$ and $\mathcal{B}_{SD}^1$, the second and third POD modes for L^2 inner product are given in Fig. 4.7. From Fig. 4.6, one can barely find any differences between these three results of φ_1. As a consequence, the corresponding ROMs for r=1 depicted in Figs. 4.8-4.9 also behave quite similar for each spatial level. One can observe in Fig. 4.8- 4.10 that the ROMs behave quite similar for each level in spite of using different scalar products.

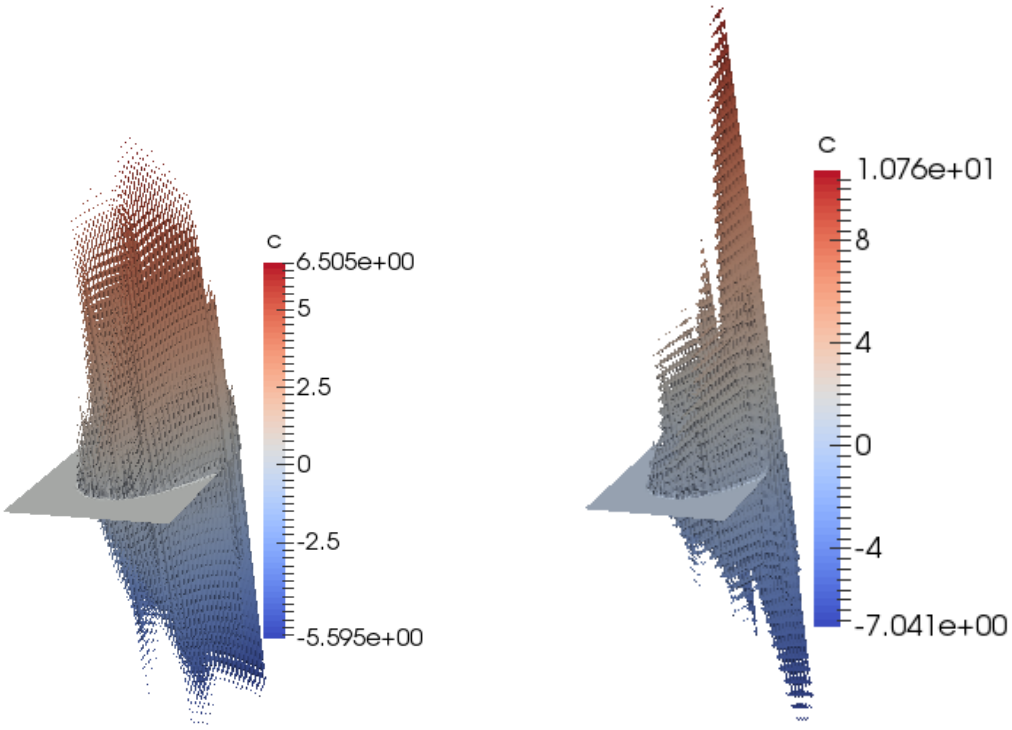


Figure 4.7: Example 4.1: $\mathcal{B}_{L^2}^2$(left), $\mathcal{B}_{L^2}^3$(right) with spatial level 7.

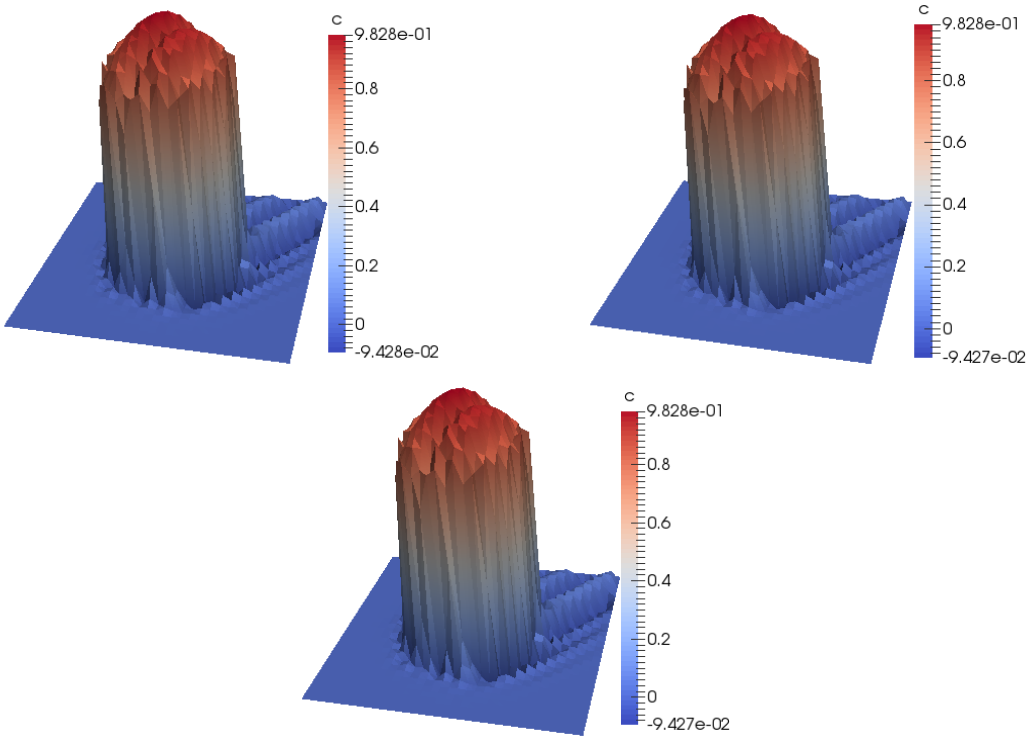


Figure 4.8: Example 4.1: $ROM_{H^1} - l5$, $ROM_{L^2} - l5$, and $ROM_{SD} - l5$ (from top left to bottom) at t=0.5 for r=1.

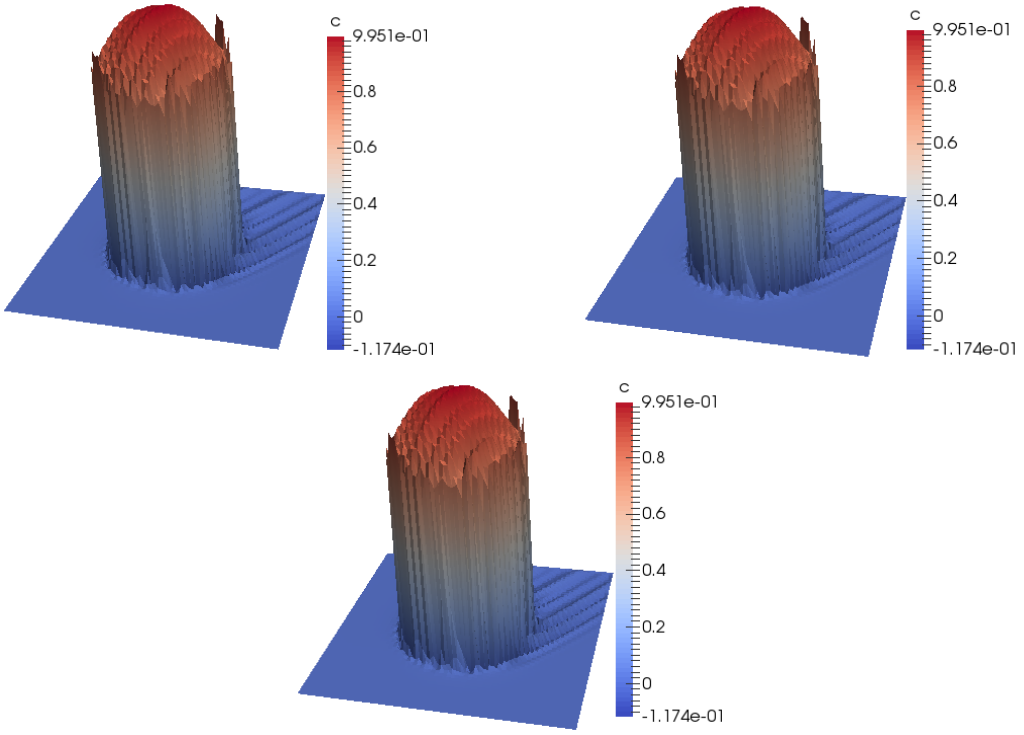


Figure 4.9: Example 4.1: $ROM(H^1) - l6$, $ROM(H^1) - l6$, and $ROM(H^1) - l6$ (from left to right) at t=0.5 for r=1.

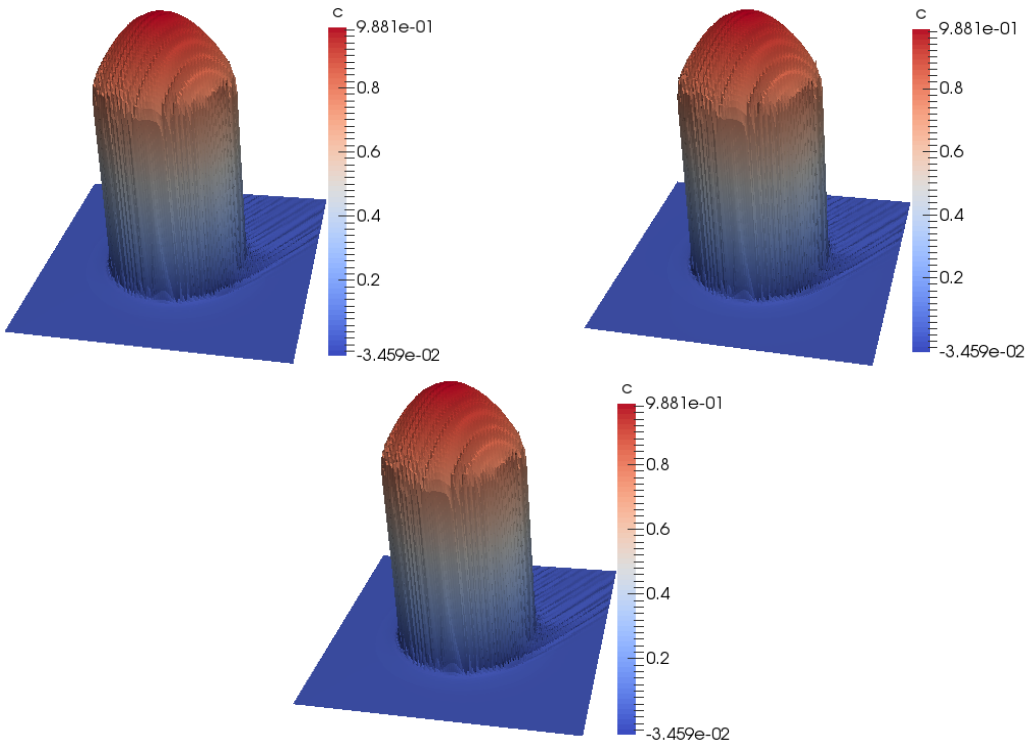


Figure 4.10: Example 4.1: $ROM_{H^1} - l7$, $ROM_{H^1} - l7$, and $ROM_{H^1} - l7$ (from left to right) at t=0.5 for r=1.

Figs. 4.11-4.12 present ROM_{L^2} for different r with spatial level 5 and 7, which will be expressed as $ROM_{L^2} - l5$ and $ROM_{L^2} - l7$, respectively. In Fig. 4.11, the $ROM_{L^2} - l5$ changes slightly with increasing r, meanwhile $ROM_{L^2} - l7$ are almost the same for different r. As a result, it seems that in this example, the influence on ROMs by different r becomes less as the grids become finer.

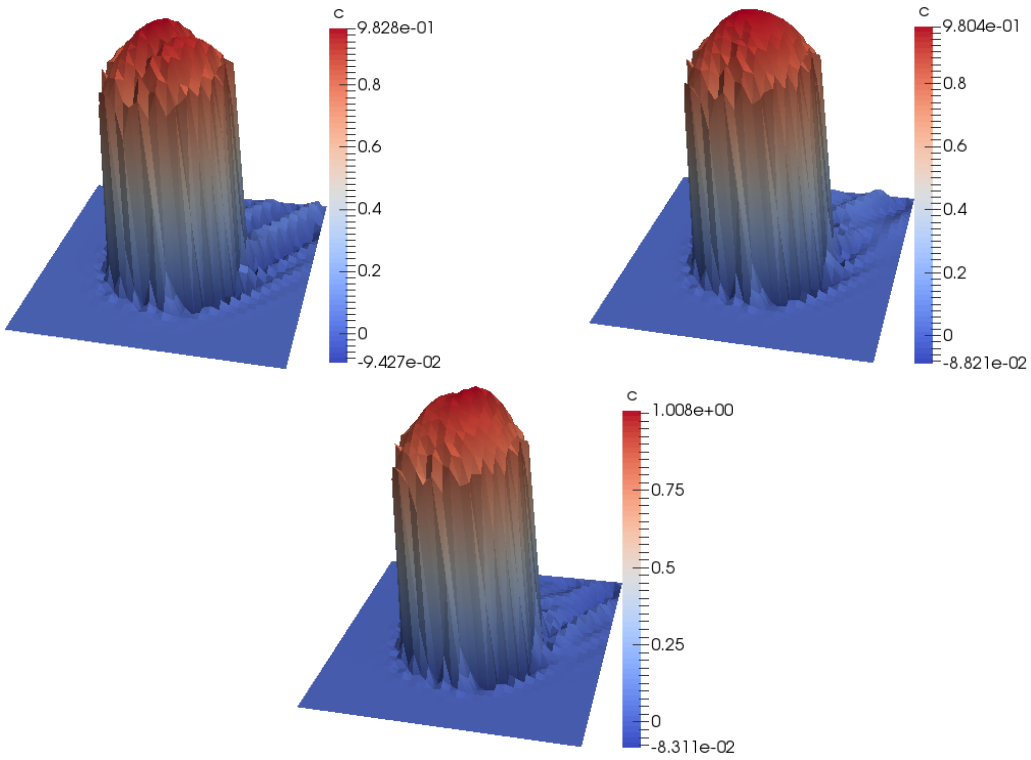


Figure 4.11: Example 4.1: ROM_{L^2} for r=1, 2 and 3 (from left to bottom) at t=0.5 for spatial level 5.

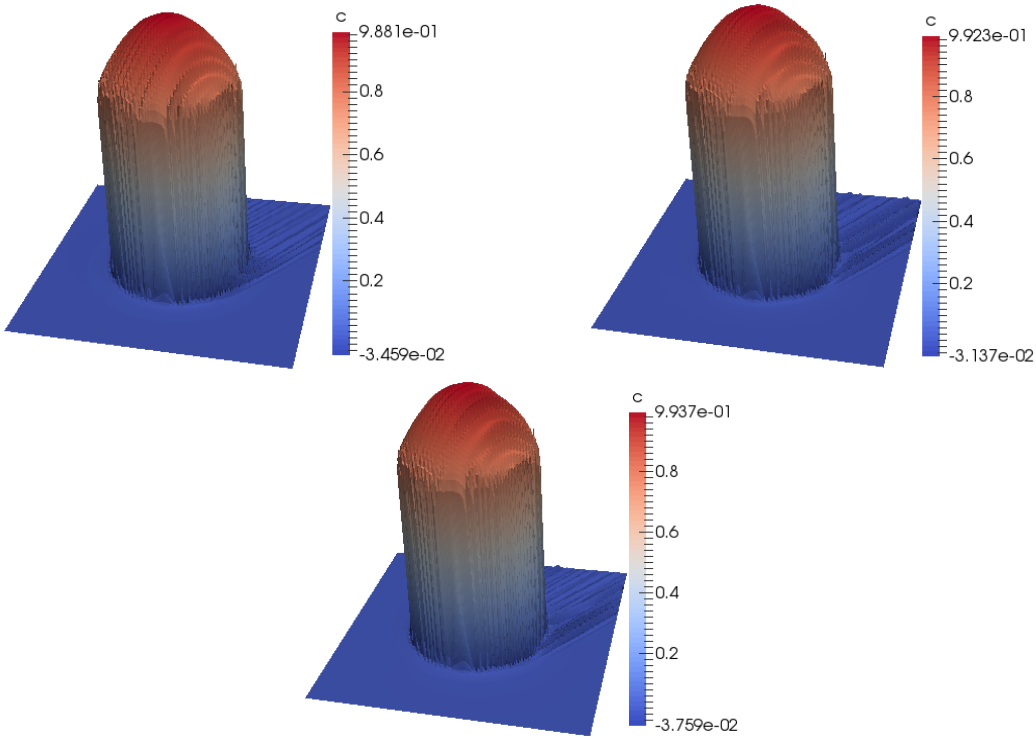


Figure 4.12: Example 4.1: ROM_{L^2} for r=1, 2 and 3 (from top left to bottom) at t=0.5 for spatial level 7.

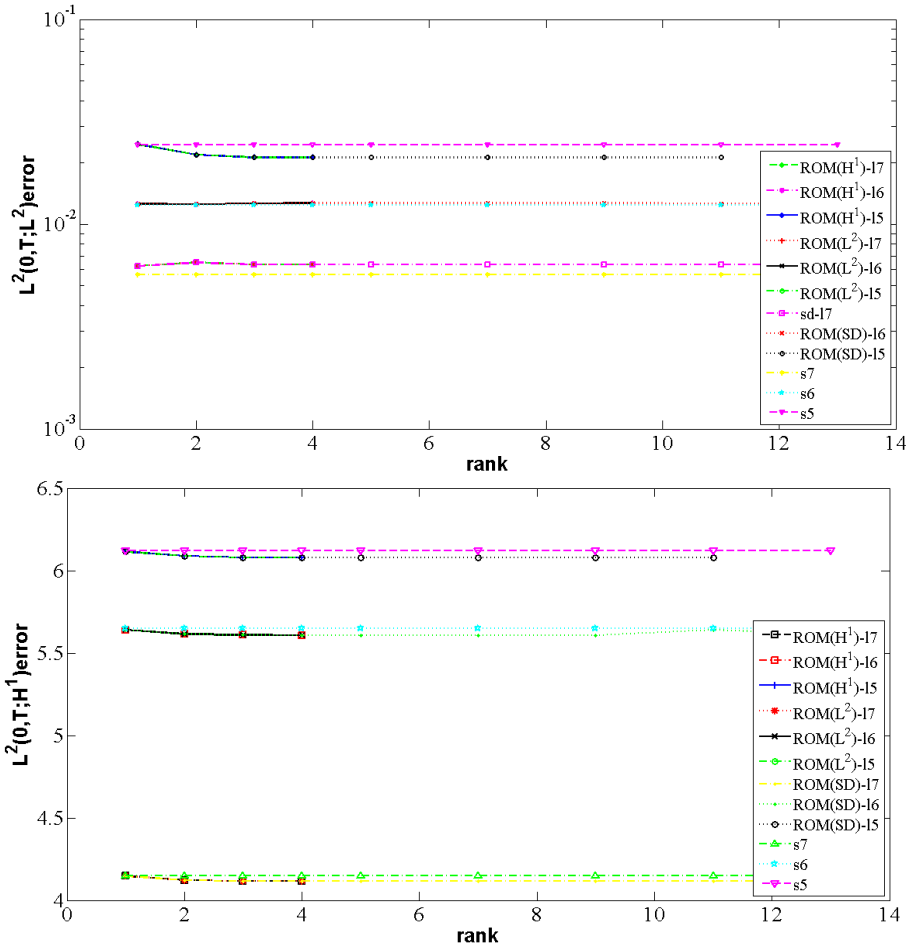


Figure 4.13: Example 4.1: $L^2(0,T;L^2(\Omega))$ (top) and $L^2(0,T;H^1(\Omega))$ (bottom) errors from ROM_{L^2} ,ROM_{H^1}, ROM_{SD} with spatial level 5, 6 and 7 and from the solutions of s5, s6, s7.

Fig. 4.13 gives the plots of errors in discrete $L^2(0,T;L^2(\Omega))$ and $L^2(0,T;H^1(\Omega))$ norms of ROMs for different inner products and for spatial levels 5, 6 and 7 with respect to r. The similar behaviours can be observed for both: the errors are reduced for both ROMs and SUPG-FEM due to the refinement of the grid. Concerning the $L^2(0,T;L^2)$ errors in the left panel, the ROMs yield more accurate results than SUPG-FEM for level 5. For level 6, the errors of ROMs are comparable to that of SUPG-FEM. Moreover, the SUPG-FEM performs better than ROMs for level 7. Whereas, observations in the right panel shows that, w.r.t the discrete $L^2(0,T;H^1)$ errors, SUPG-FEM always performs better than ROMs for all spatial levels. See in Figs. 4.14-4.15 the corresponding temporal evolution of errors for r=3 in both norm. For $r \leqslant 3$, the errors of ROMs in $L^2(0,T;H^1)$ norm decrease slightly with increasing r for each level, while such a behavior appears only for level 5 in $L^2(0,T;L^2)$ norm. In spite of different scalar products, some similar behaviors can be observed for both error norms: firstly, the errors are reduced for both SUPG-FEM and ROMs due to the refinement of the grid. Secondly, the impact of different inner products is negligible for each spatial level. This is in accordance with the observations in Fig. 4.6 for $\mathcal{B}_{1L^2}$, $\mathcal{B}_{1H^1}$, $\mathcal{B}_{1SD}$ and Figs. 4.8- 4.10 for ROM_{L^2}, ROM_{H^1}, ROM_{SD}.

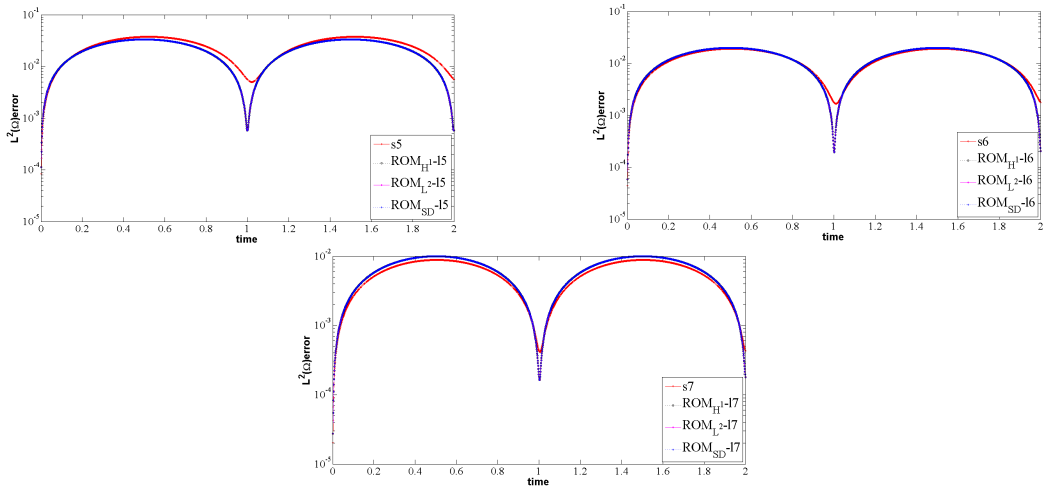


Figure 4.14: Example 4.1: $L^2(\Omega)$ error from SUPG solution and ROM_{L^2}, ROM_{H^1} and ROM_{SUPG} with spatial levels 5 (top left), 6 (top right) and 7 (bottom) for r=3.

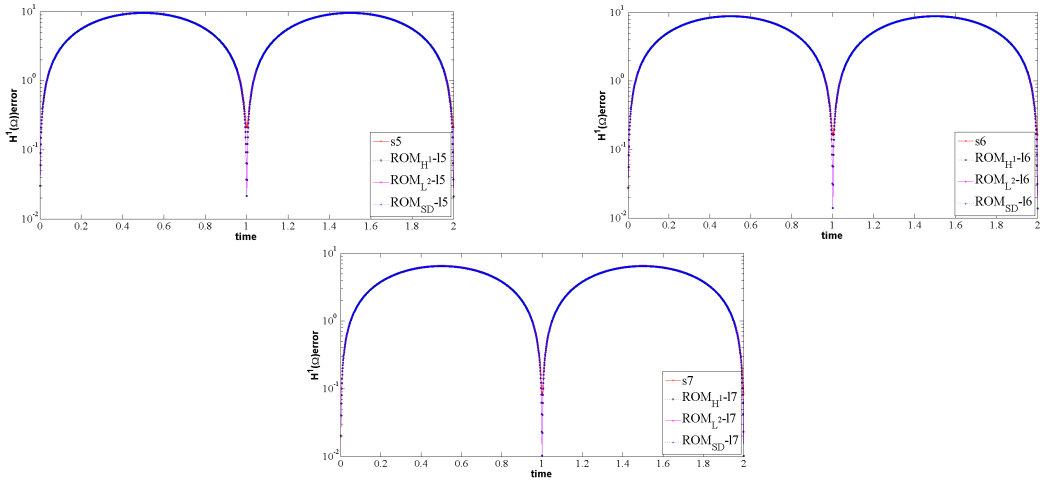


Figure 4.15: Example 4.1: $H^1(\Omega)$ error from SUPG solution and ROM_{L^2}, ROM_{H^1} and ROM_{SD} with spatial levels 5 (top left), 6 (top right) and 7 (bottom) for r=3.

4.2 Traveling wave for non-convection-dominated case

This example is taken from [7]. It is given in the domain $\Omega = (0,1)^2$ and (0,T)=(0,1). The coefficients in (3.1) are chosen to be $\varepsilon = 10^{-4}$, $\mathbf{b} = (\cos(\pi/3), \sin(\pi/3))^T$, and c=1.

Similar to Example 4.1, this problem has a prescribed solution, which is defined by

$$u(t,x,y) = 0.5\sin(\pi x)\sin(\pi y)\left[\tanh(\frac{x+y-t-0.5}{\sqrt{\varepsilon}})+1\right]. \tag{4.2}$$

The right-hand side f, the initial condition u_0, and the boundary condition were chosen such that (4.2) satisfies the boundary value problem, see [6].

In this example, mainly the G-FEM was applied for computing the snapshots and also the construction of the ROMs, and the L^2 inner product is applied for the computation of the POD modes and ROMs. The first three POD modes from different amount of snapshots will be presented. In addition, the sensitivity of the ROMs with respect to the rank of the POD modes used for building the ROMs and the amount of snapshots fetched from the finite element solution will be explored. Finally, the corresponding error behavior will be studied and discussed.

The backward Euler scheme was applied for temporal discretization with the length of time step $\Delta t = 10^{-3}$. All the test problems were defined in the unit square. For the coarsest grid (call it 'level 0'), the unit square was divided by diagonal from bottom left to top right into two triangles. For the construction of the further finer grids, the uniform grid refinement was applied. Here in this example, snapshots were computed on level 7, i.e., $h = 1.1 \cdot 10^{-2}$, with 16641 degrees of freedom. This example is not a convection-dominated problem, since the layer width $\sqrt{\varepsilon} = 10^{-2}$ is of similar size as the grid size, which implies that the grid is sufficiently fine to resolve the layer. Hence the application of stabilization method is not necessary, and G-FEM suffices to give non-oscillating solutions. This is verified by Fig. 4.16. To obtain the snapshots, the Galerkin conforming piecewise (P1) finite element method was used.

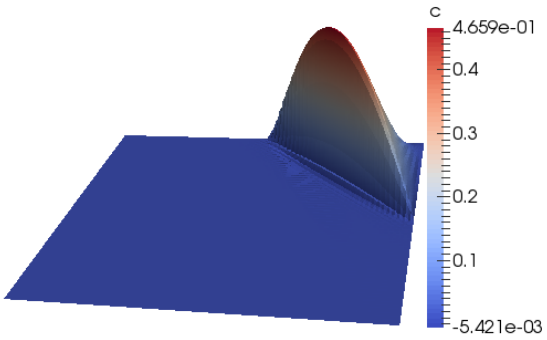


Figure 4.16: Example 4.2: the solution at t=0.1 for G-FEM.

To investigate the sensitivity of the numerical results of the reduced-order model with respect to the different amount of snapshots, every twice, fourth, tenth, twentieth, fiftieth, hundredth and the full solutions were stored. By this means, the snapshots are fetched evenly from the finite element solution (since the total amount of the full solutions are

divided evenly by these numbers). Moreover, the initial solution is always included in every snapshot set. Therefore 11, 21, 51, 101, 251, 501 and 1001 snapshots were used for the computation of snapshots, respectively.

Number of snapshots	11	21	51	101	251	501	1001
Rank of POD modes	10	20	50	100	131	131	132

Table 4.1: Example 4.2: Ranks of POD modes for 11, 21, 51, 101, 251, 501, and 1001 snapshots (from left to right).

Fig. 4.18- 4.24 show the pictures of the snapshot mean value given by (2.21) and the first three POD modes φ_1, φ_2 and φ_3, for 11, 21, 51, 101, 251, 501, and 1001 snapshots, respectively. For simplification, the i-th POD mode using different amount of snapshots will be denoted as $\mathcal{B}^i(n)$, and the ROMs using different amount of snapshots will be expressed as $ROM(m)$, $m \in \{11, 21, 51, 101, 251, 501, 1001\}$. Tab. 4.1 presents the dimension of these 7 snapshot spaces. Define the maximal value of the dimensions among all the snapshot spaces as threshold rank. It shows that the dimension of the snapshot space V^h increases as the amount of snapshots increases until the threshold rank has been reached. Fig. 4.17 gives the plot of eigenvalues corresponding to the POD modes for each snapshot space. Note that the eigenvalues practically decrease exponentially with the rank of POD modes increasing, which implies that the eigenvalues reduce rapidly, especially for the first and the second eigenvalues such that the former one is significantly larger than the second one.

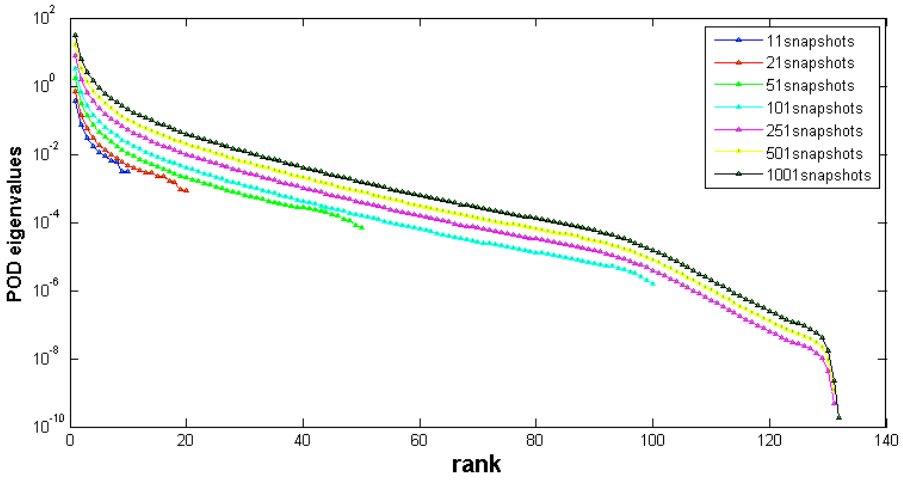


Figure 4.17: Example 4.2: POD eigenvalues for 11, 21, 51, 101, 251, 501, and 1001 snapshots.

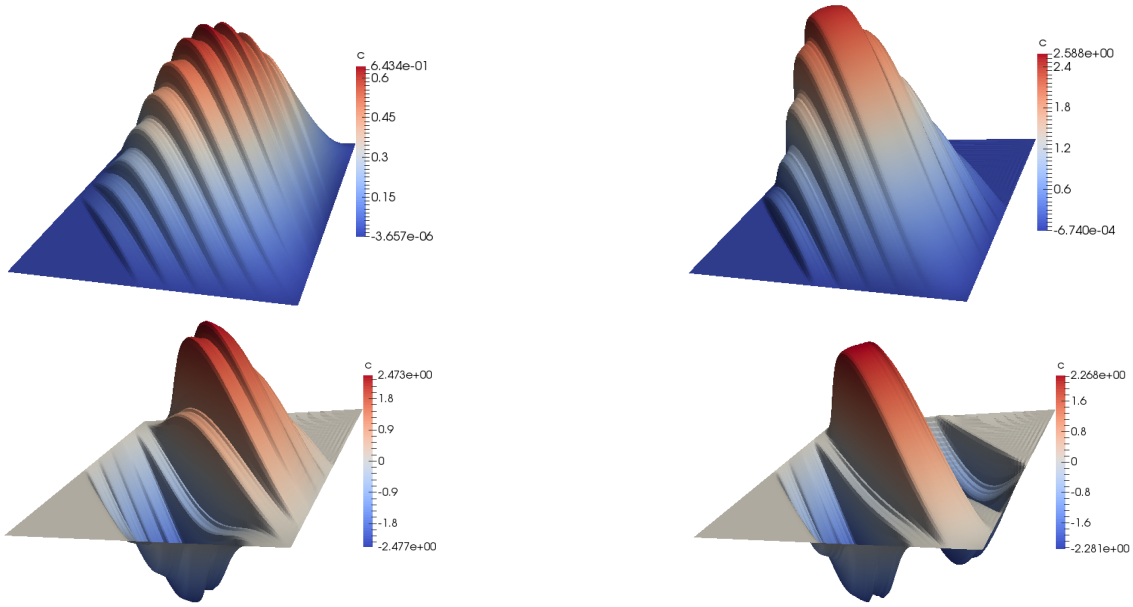


Figure 4.18: Example 4.2: snapshot mean value and $\mathcal{B}^1(11)$, $\mathcal{B}^2(11)$, and $\mathcal{B}^3(11)$.

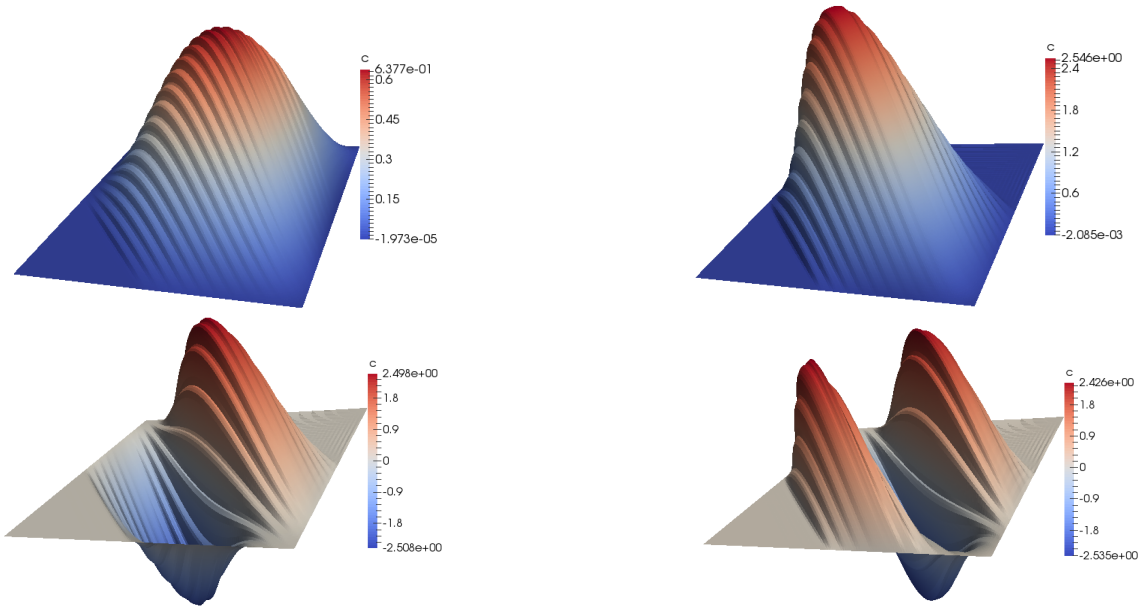


Figure 4.19: Example 4.2: snapshot mean value and $\mathcal{B}^1(21)$, $\mathcal{B}^2(21)$, and $\mathcal{B}^3(21)$.

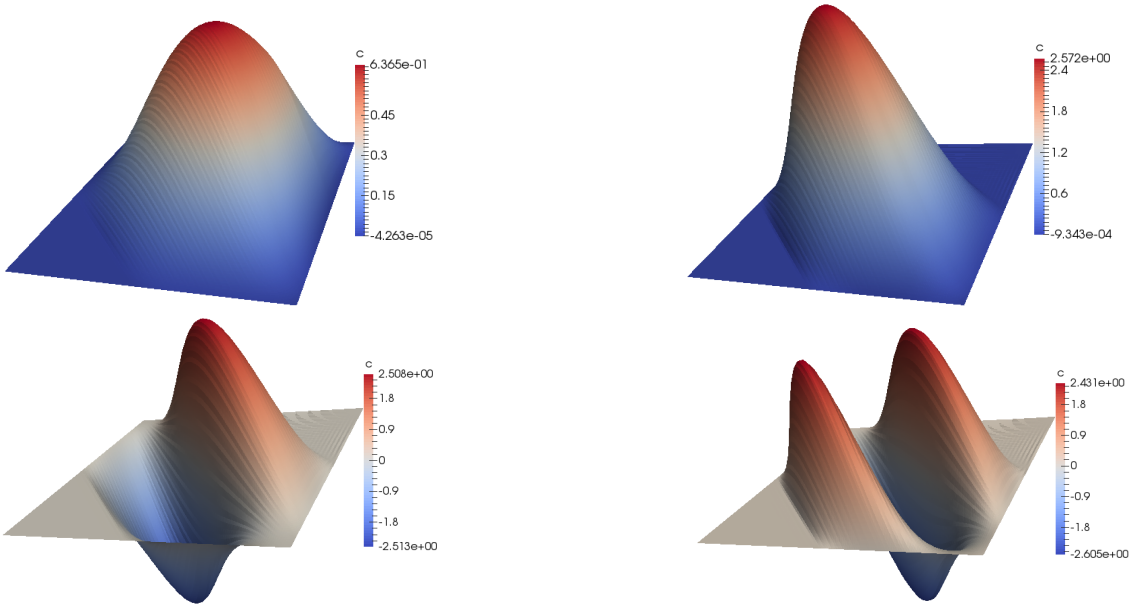


Figure 4.20: Example 4.2: snapshot mean value and $\mathcal{B}^1(51)$, $\mathcal{B}^2(51)$, and $\mathcal{B}^3(51)$.

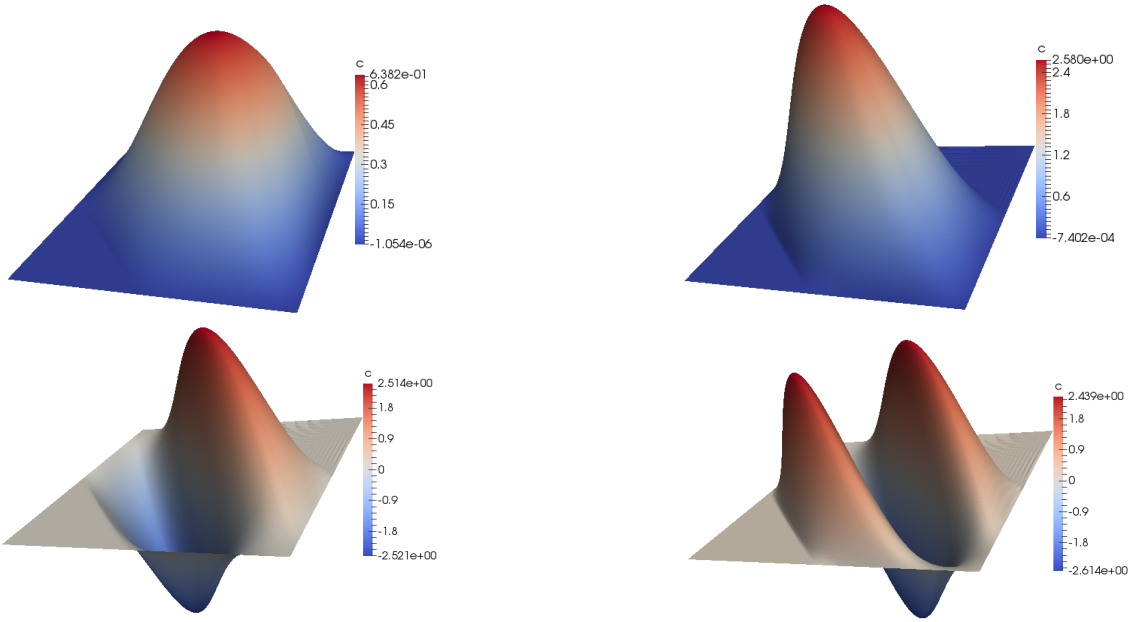


Figure 4.21: Example 4.2: snapshot mean value and $\mathcal{B}^1(101)$, $\mathcal{B}^2(101)$, and $\mathcal{B}^3(101)$.

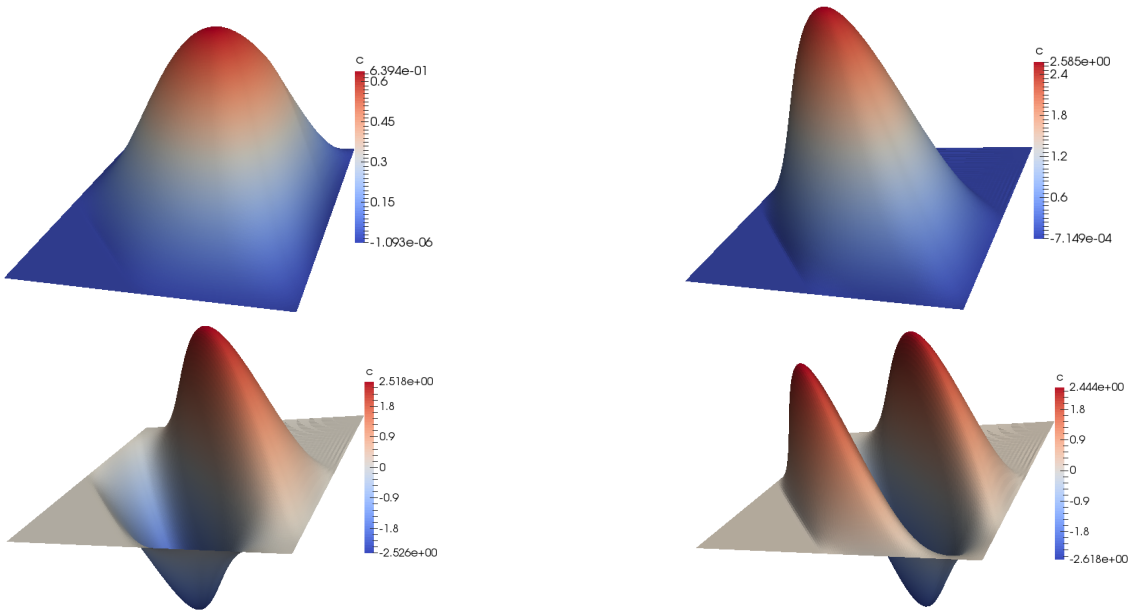


Figure 4.22: Example 4.2: snapshot mean value and $\mathcal{B}^1(251)$, $\mathcal{B}^2(251)$, and $\mathcal{B}^3(251)$.

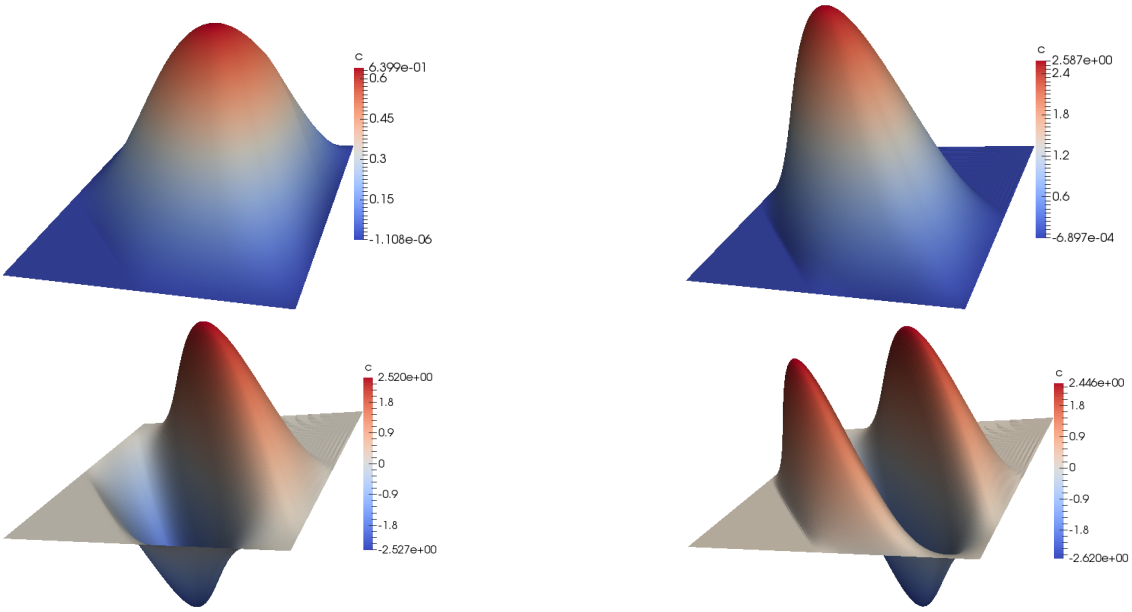


Figure 4.23: Example 4.2: snapshot mean value and $\mathcal{B}^1(501)$, $\mathcal{B}^2(501)$, and $\mathcal{B}^3(501)$.

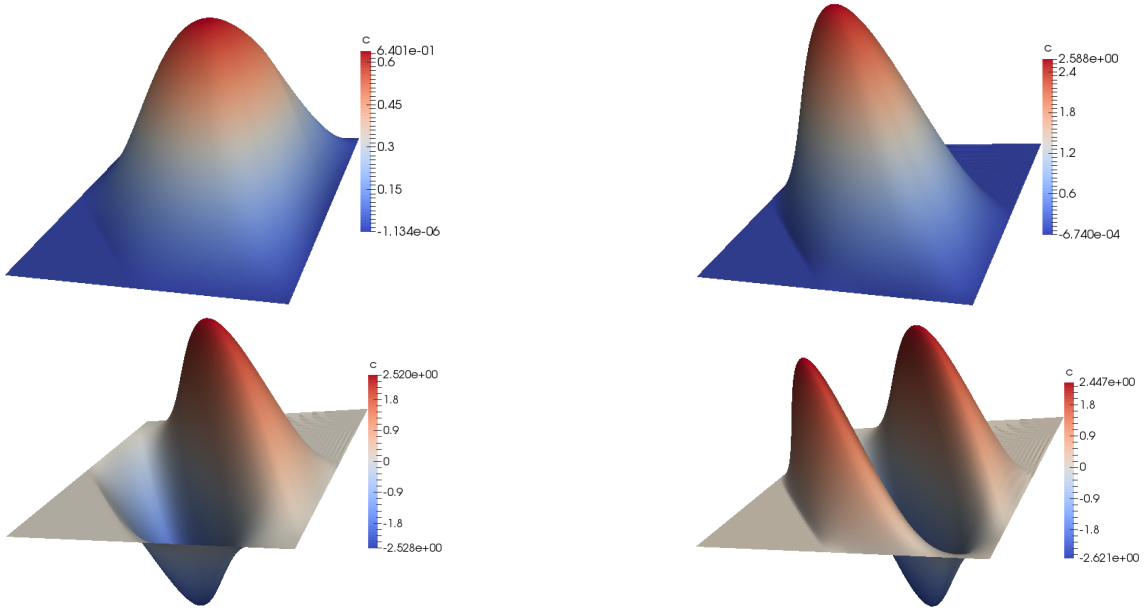


Figure 4.24: Example 4.2: Snapshot mean value and $\mathcal{B}^1(1001)$, $\mathcal{B}^2(1001)$, and $\mathcal{B}^3(1001)$ (from top to bottom).

In Figs. 4.18- 4.21, the pictures of the snapshot mean value and $\mathcal{B}^1(m)$, $\mathcal{B}^2(m)$ and $\mathcal{B}^3(m)$, $m \in \{11, 21, 51\}$ are getting smoother and show more details as the amount of snapshots is becoming larger. Nevertheless, the pictures in Fig. 4.21- 4.24 show quite similar results when the amount of snapshots is equal or bigger than 101.

In Fig. 4.25, the ROM(1001)s at t=1.0 are depicted for different r. It is observed that as r become higher, the size of spurious oscillations of ROMs decreases. For r=100, the performance of ROMs becomes very close to the solution of G-FEM in Fig. 4.16. The similar results are also obtained for 101, 251 and 501 snapshot spaces, and the corresponding pictures won't be shown here for the sake of brevity.

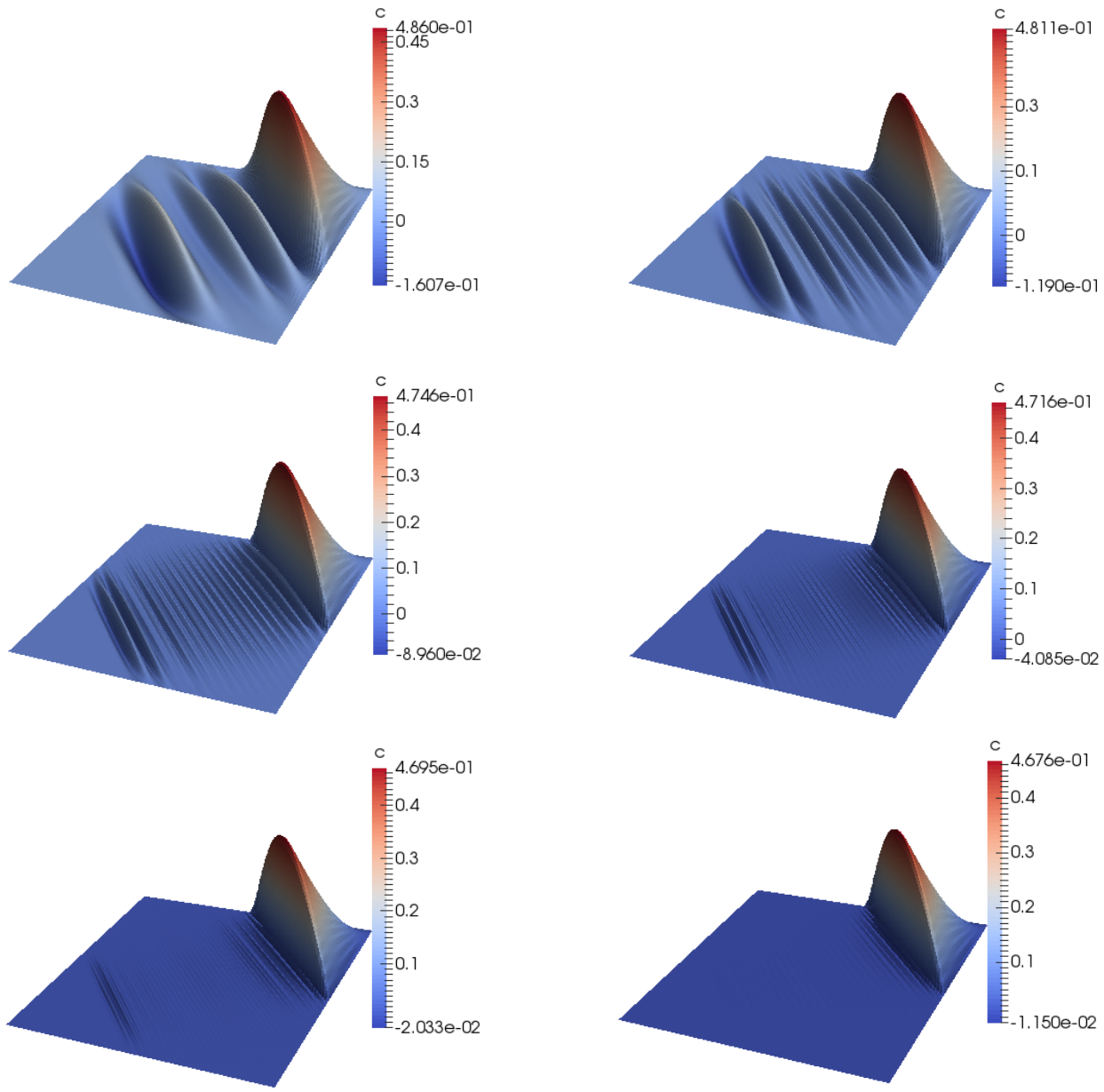


Figure 4.25: Example 4.2: $ROM(1001)s$ at t=1.0 for r=10, 20, 40, 60, 80, 100 (from top left to bottom right).

It is already known that the POD basis is computed from the snapshot spaces. To study the impact of different snapshot spaces on the resulting ROMs, fixed values of r is considered for different snapshot spaces. For r=50, concerning the spurious oscillations, $ROM(101)$, $ROM(251)$, $ROM(501)$ are similar, and perform slightly better than $ROM(51)$. Meanwhile, see in Fig. 4.27 for low r, e.g., r equal to 20, despite the fact that there are always big spurious oscillations on the plane for each snapshot space, the ROMs for snapshots equal or larger than 51 yield much better results than $ROM(21)$. Yet the differences between ROM(m)s, $m \in \{51, 101, 251, 501, 1001\}$ are negligible.

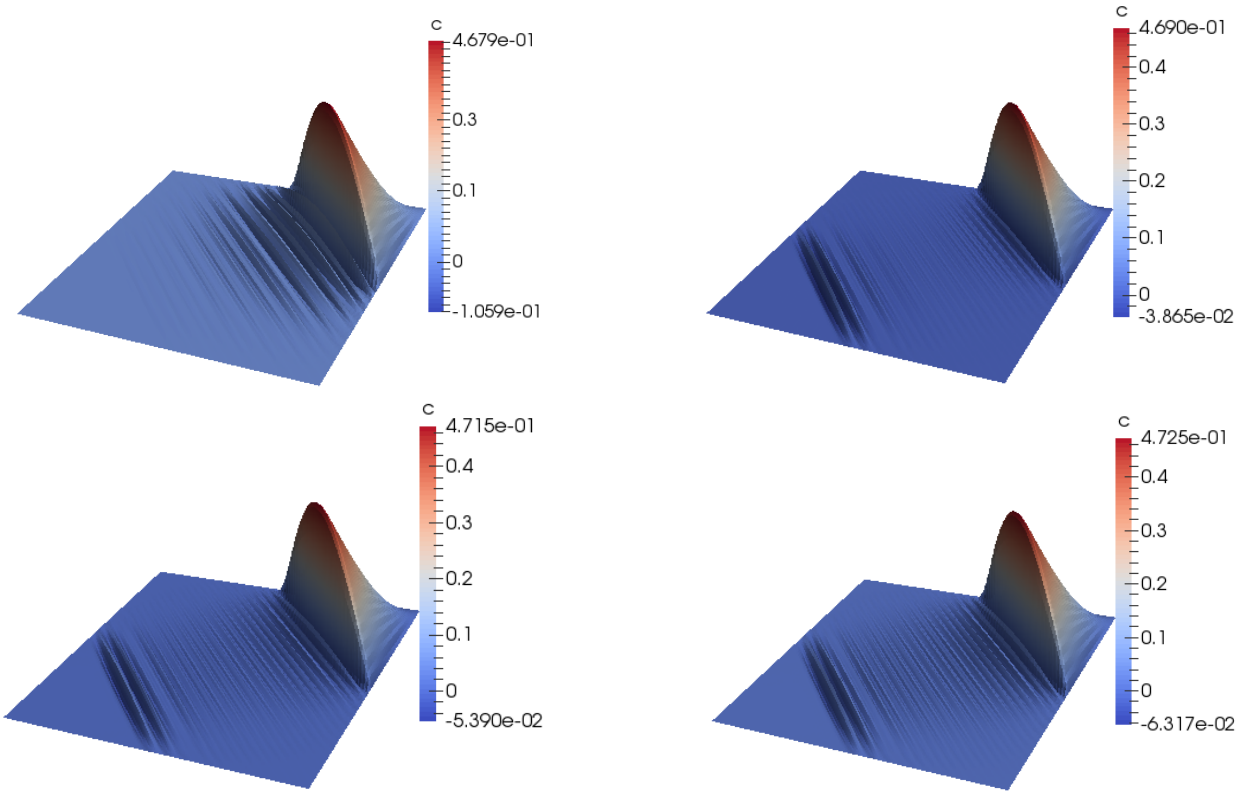


Figure 4.26: Example 4.2: $ROM(m)$, $m \in \{51, 101, 251, 501\}$ at t=0.5 (from top left to bottom right) for r=50.

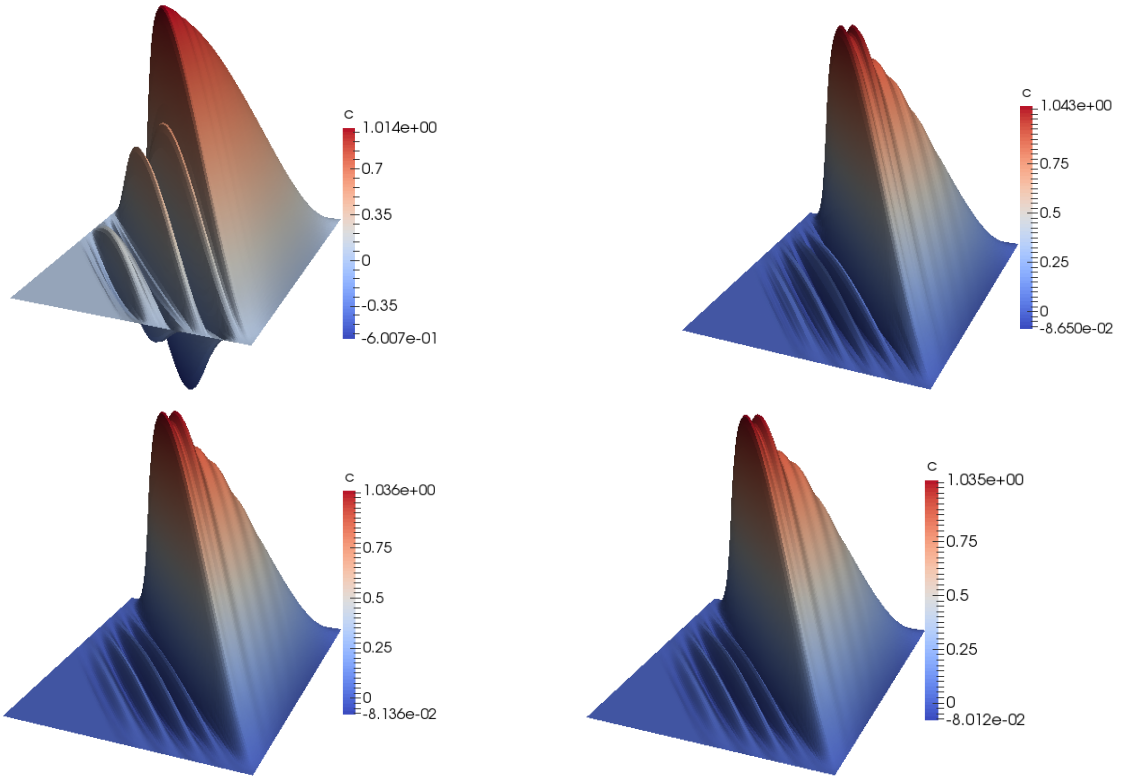


Figure 4.27: Example 4.2: $ROM(m)$, $m \in \{21, 51, 101, 251\}$ at t=1.0 (from top left to bottom) for r=20.

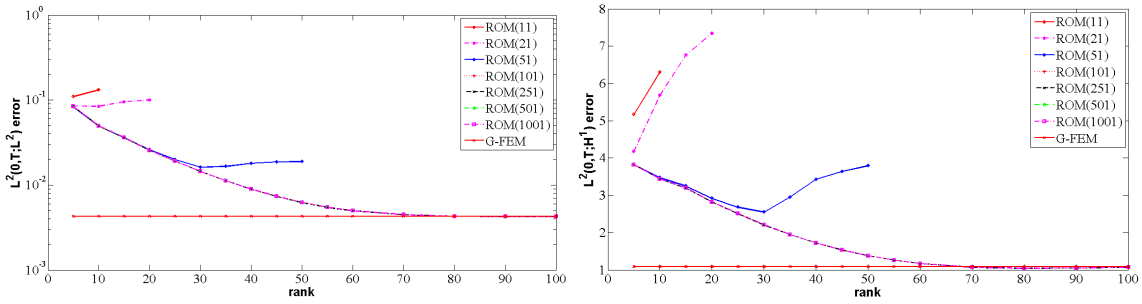


Figure 4.28: Example 4.2: $L^2(0,T;L^2)$ (left) and $L^2(0,T;H^1)$ (right) errors for different snapshots.

The plots in Fig. 4.28 present the resulting errors in discrete $L^2(0,T;L^2(\Omega))$ norm and the $L^2(0,T;H^1(\Omega))$ norm. It is observed that, concerning both kinds of errors, the accurate ROM in this example is achieved for $m \geqslant 101$ and $r \geqslant 100$. In the left plot for $L^2(0,T;L^2)$ errors, the errors for $ROM(11)$ and $ROM(21)$, which are much larger than the errors for the rest ROMs, increase as the rank of POD modes become higher. For the snapshots which are equal or larger than 51, concerning the errors, even though the amount of snapshots increases, which implies that the behaviour of the ROMs remains unchanged. In addition, as r increases, the errors for these ROMs keep decreasing, except for the case of $ROM(51)$ for r$\geqslant$30, the errors of which stay almost the same. For r from 5 to 10, the errors damp fastest, and with r being higher, the damping speed becomes slower. The reason for this behavior can be achieved by observations in Fig. 4.17 that

since the eigenvalues decrease rapidly, the information of data given by the corresponding POD modes decrease fast based on (2.14). For r is equal or higher than 60, the accuracies of the ROM(m)s, m$\in \{101, 251, 501, 1001\}$ are getting closer to the one by G-FEM, but never exceed it. This behavior is in contrast to the case of discrete $L^2(0,T;H^1)$ norm, in which the errors caused by ROMs are less than the ones by G-FEM for r$\geqslant$70.

4.3 Traveling wave for convection-dominated case

The studied equation in this section share the same coefficients and conditions as in Example 4.2, except that the constant diffusion coefficient is set to $\varepsilon = 10^{-6}$. And the way to choose snapshots is the same to Example 4.2 that 11, 21, 51, 101, 251, 501 and 1001 snapshots were stored for the computation of different POD bases. What's more, for the computation of snapshots and the construction of ROMs, P1 finite element method was used on spatial level 7, and the L^2 inner product is applied for the computation of the POD modes and ROMs. For the time discretization, the backward Euler method was considered on equidistant time intervals with $\Delta t = 10^{-3}$.

Example 4.3.1. *Applying SUPG-FEM for the computation of snapshots.*

In this example, the SUPG-FEM was applied for the computation of snapshots, and SUPG stabilized ROMs (SUPG-ROM) were studied. Besides, the comparisons of applying the Galerkin method and the SUPG method will be presented.

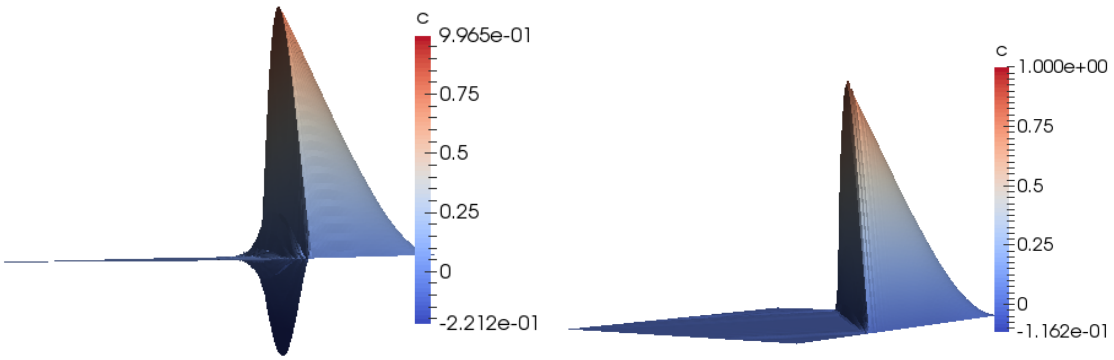


Figure 4.29: Example 4.3.1: Solution at t=0.8 by G-FEM (left) and SUPG-FEM (right).

The solution of this problem by G-FEM at t=0.8 is shown in the left panel of Fig. 4.29. It can be seen that there are large spurious oscillations, especially for the area below the plane. The explanation of this behavior comes from the fact that the problem becomes a convection-dominated problem due to the reduced value of ε. Hence the application of a suitable stabilization method is needed. The right plot of Fig. 4.29 presents the solution by SUPG-FEM, which seems to be much smoother and reasonable.
The snapshots are computed by SUPG-FEM for the reason discussed above. In Tab. 4.2, the ranks of the POD basis for each snapshot-space are shown. Different from Example 4.2 for $\varepsilon = 10^{-4}$, there exits no threshold rank for these 7 snapshot spaces. As the number of snapshots increases, the rank increases as well. In particular, for 501 and 1001 snapshots, the ranks are even twice higher than the corresponding values in Example 4.2, see Tab. 4.1. In practice, the magnitude of r reflects the complexity of the problem to some extent. This seems to be due to the small ε that even though the stabilization method was used, there are always more oscillations than in the non-convection-dominated problem in Example 4.2. Hence Example 4.3.1 is considered as a more complicated problem than Example 4.2.

Number of snapshots	11	21	51	101	251	501	1001
Rank of POD modes	10	20	50	100	246	352	376

Table 4.2: Example 4.3.1: Ranks of POD modes for 11, 21, 51, 101, 251, 501 and 1001 snapshots (from left to right).

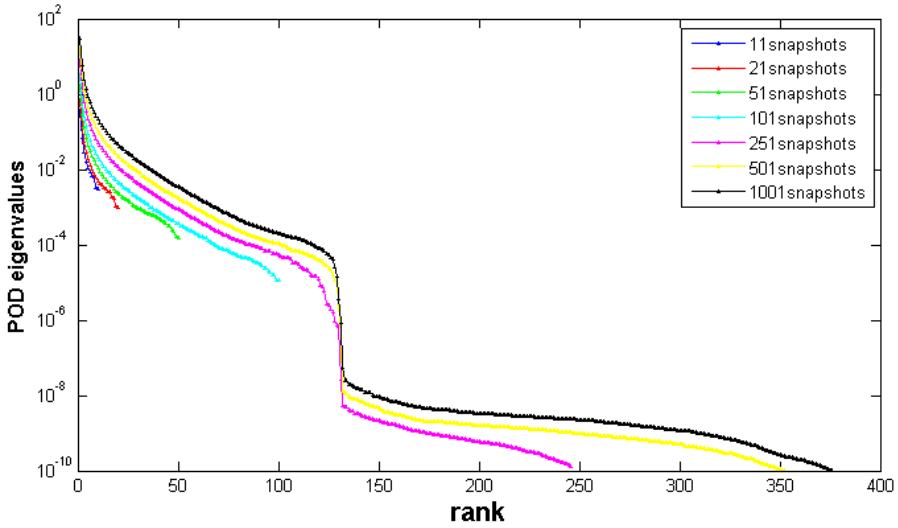


Figure 4.30: Example 4.3.1: POD eigenvalues for 11, 21, 51, 101, 251, 501 and 1001 snapshots.

Fig. 4.30 shows the eigenvalues for different snapshot spaces. For 251, 501, and 1001 snapshots, the total amounts of eigenvalues are much bigger than the corresponding ones in Tab. 4.1 in Example 4.2. Meanwhile, similar as that in Example 4.2 (see in Fig. 4.30), there is also a steep drop around r≈130. Hence the extra POD modes for snapshots from SUPG-FEM may come from the numerical artefacts, since although the SUPG-FEM is used for stabilization, there are still spurious oscillations in the solution, which therefore come into the snapshots.
The comparisons of the eigenvalues for case $\varepsilon = 10^{-4}$ in Example 4.2 and $\varepsilon = 10^{-6}$ in this example are given in Fig. 4.31 for r⩽150. For very low r, e.g., $r \leqslant 15$, the eigenvalues for both are quite similar. As r becomes larger, the eigenvalues for $\varepsilon = 10^{-4}$ damp faster than the corresponding ones for $\varepsilon = 10^{-6}$ for each snapshot space. Notice that there is a steep drop around r≈130, which is similar as that in Example 4.2.

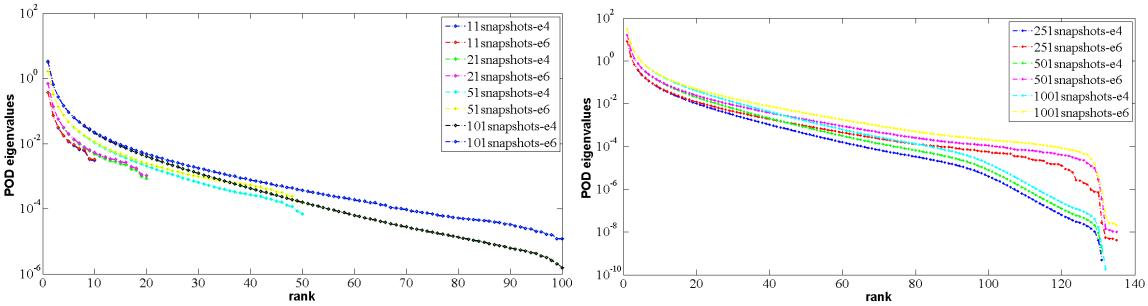


Figure 4.31: Example 4.3.1: Comparisons of eigenvalues between $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-6}$.

To distinguish the first POD modes B^1 from Example. 4.2, the first POD modes from snapshots obtained by SUPG-FEM will be denoted as $SUPG - B^1$. Fig. 4.32 offers the first POD modes from $21, \ldots, 1001$ snapshots. Similar behavior as that in Example 4.2 is observed: the POD modes are getting smoother as the amount of snapshots increases. For the amount of snapshots $m \geqslant 251$, it seems that the POD modes achieve its highest accuracy, and remain unchanged despite of the increase of m.

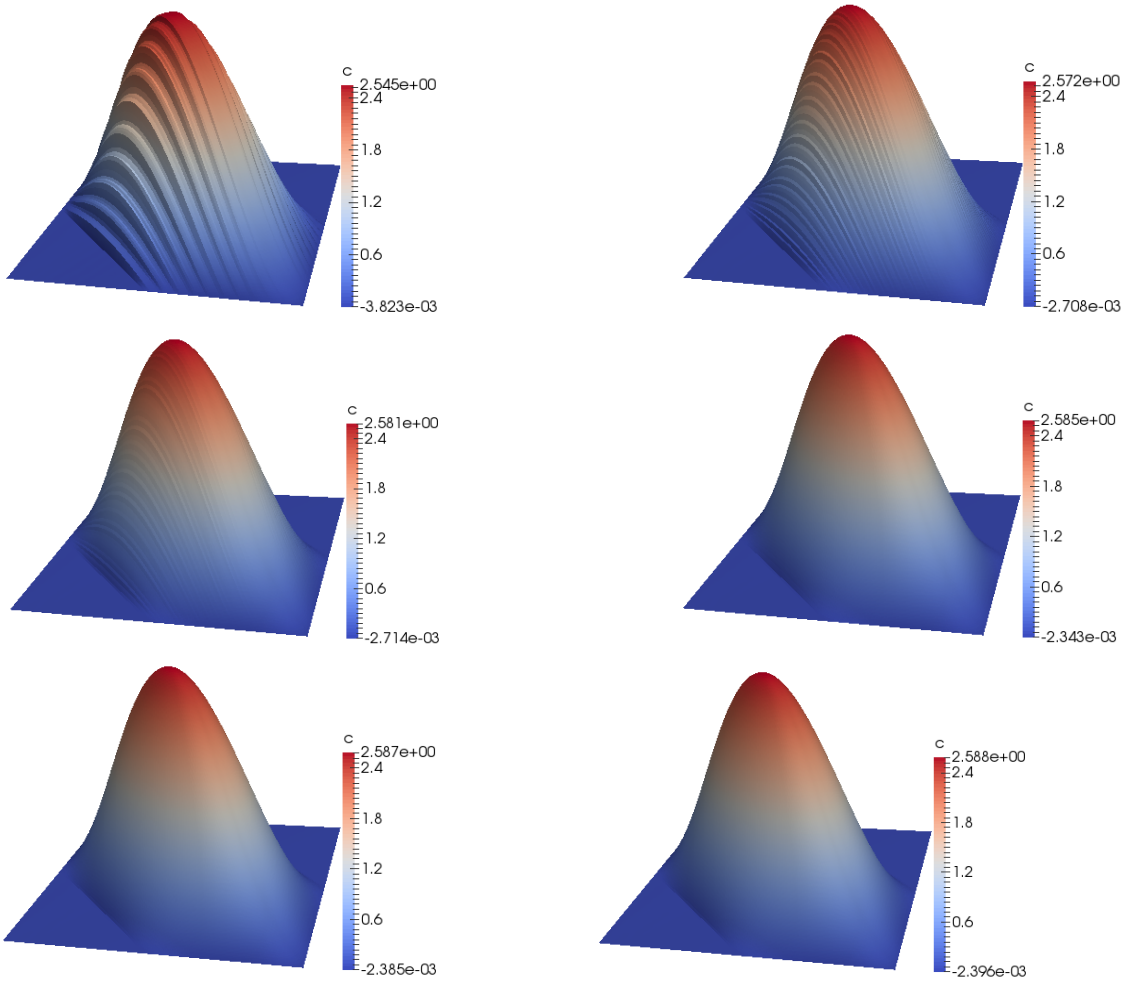


Figure 4.32: Example 4.3.1: $SUPG - \mathcal{B}^1(m)$, $m \in \{21, 51, 101, 251, 501, 1001\}$(from top left to bottom right).

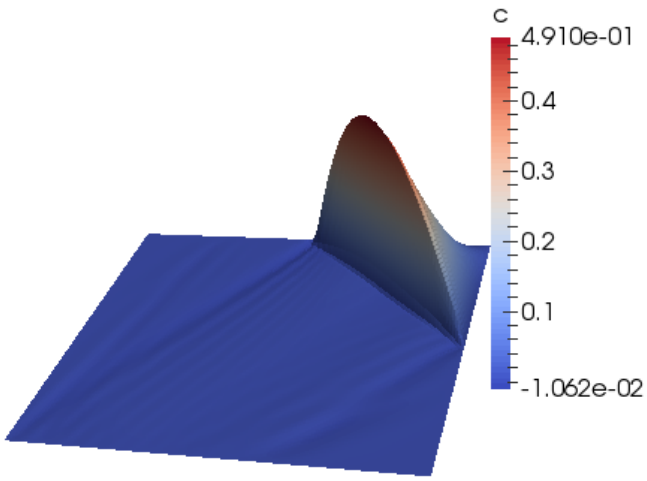


Figure 4.33: Example 4.3.1: Solution of SUPG-FEM at t=1.0.

For the goal of investigating the sensitivity of the ROMs with respect to r, like in Example 4.2, $ROM(1001)$ for different r are shown in Fig. 4.34. One gets some similar observations to the case $\varepsilon = 10^{-4}$ using G-FEM: as r increases, the oscillations decrease, and the ROMs perform better. For r$\geqslant$60, the ROMs behave very similar, and the result for r=150 is quite close to the solution of the SUPG-FEM, see Fig. 4.33. In addition, it is worth mentioning that even though the SUPG-FEM was applied for stabilization, some small oscillations always exist.

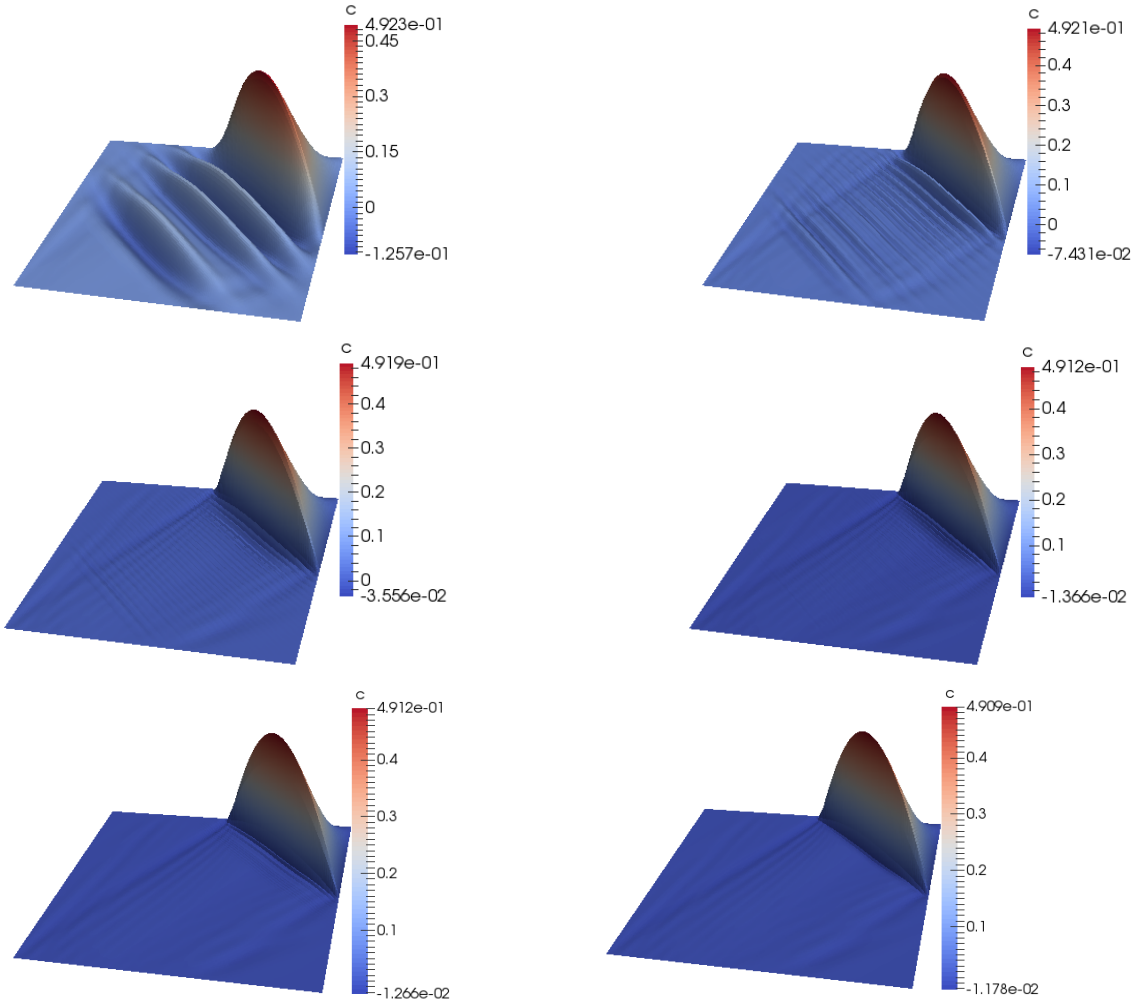


Figure 4.34: Example 4.3.1: $ROM(1001)$ at t=1.0 for r=10, 30, 60, 90, 120, 150 (from top left to bottom right).

To investigate how the amount of snapshots effects the ROMs, the ROMs of $r = 20, 50$ using different amount of snapshots are considered, as in Example. 4.2. Unlike the situation in Example 4.2 for $\varepsilon = 10^{-4}$, the $ROM(21)$ in Fig. 4.35 for r=20 obtains much less spurious oscillations than the corresponding one in Fig. 4.27. This may be due to the usage of the stabilization method. When comparing the pictures in Fig. 4.36 (representing the ROMs for r=50), it is observed that $ROM(101)$, $ROM(251)$ and $ROM(501)$ perform quite similar, and only slightly better than $ROM(51)$. It seems that, for fixed r, the ROMs from big amount of snapshots show better behavior than the ROMs from small amount of snapshots, until the amount of snapshots becomes big enough (in this case, it is $m = 101$).

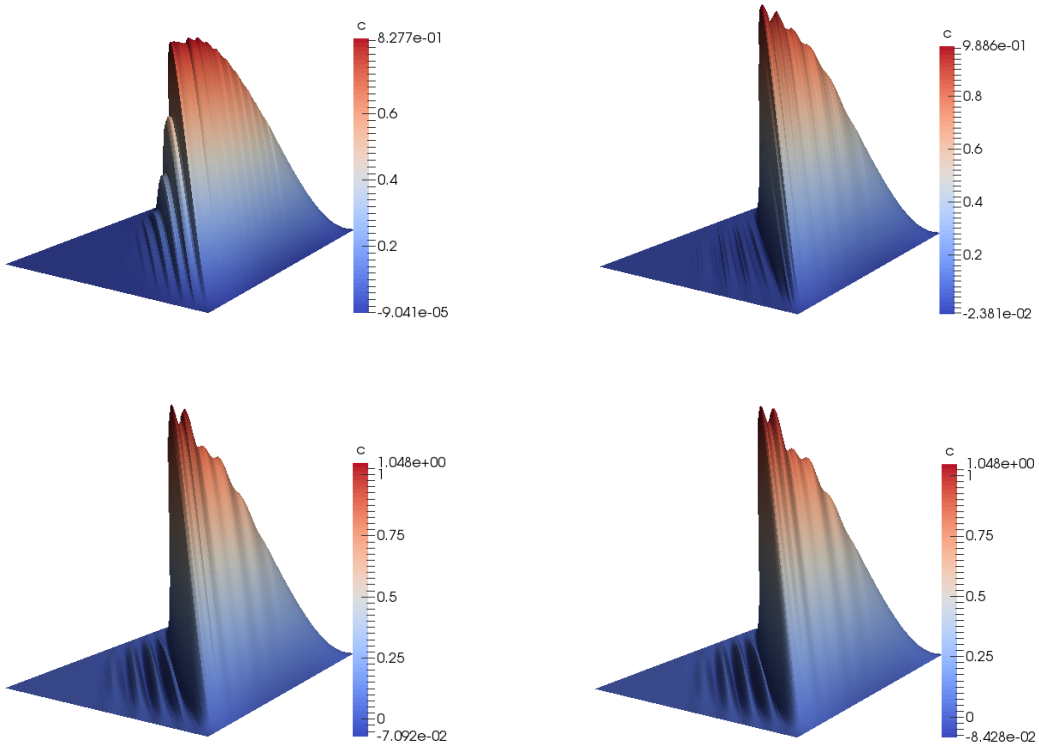


Figure 4.35: Example 4.3.1: $ROM(m)$, $m \in \{21, 51, 101, 251\}$ (from top left to bottom right) for r=20.

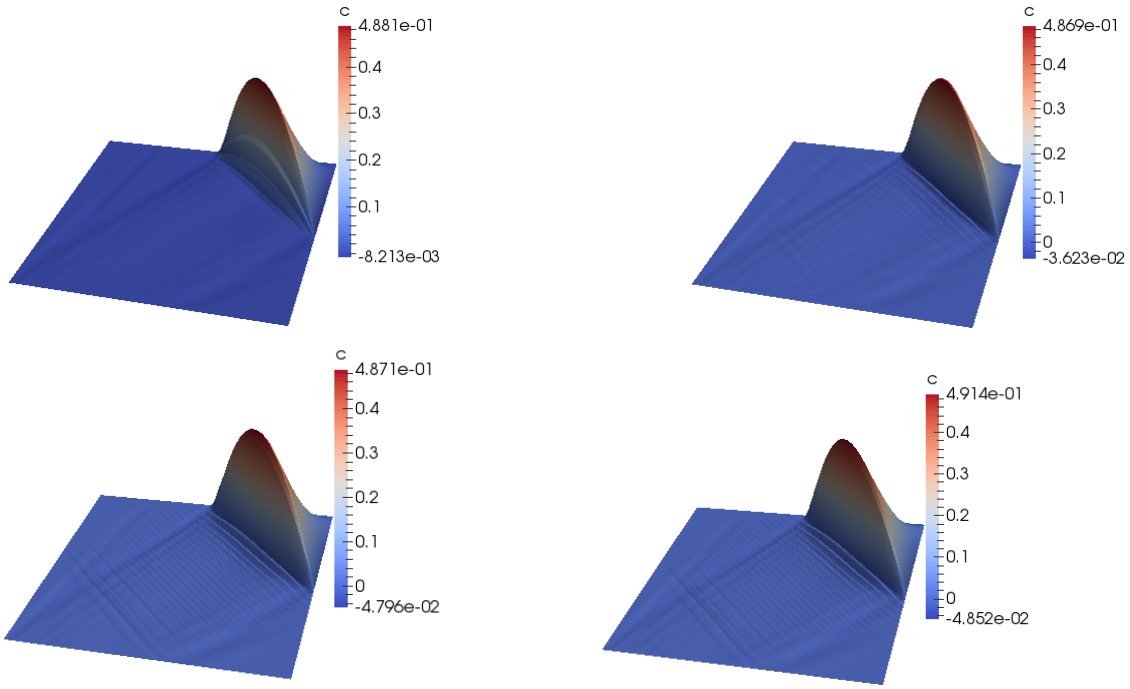


Figure 4.36: Example 4.3.1: $ROM(m)$, $m \in \{51, 101, 251, 501\}$ (from top left to bottom right)at t=1.0 for r=50.

In Fig. 4.37 the discrete errors in $L^2(0,T;L^2)$ and $L^2(0,T;H^1)$ norm are shown. From

the plots of both errors, it suggests that the accurate ROM is obtained for $m \geqslant 251$ and $r \geqslant 150$. For ROMs from all snapshot spaces, the increase of r generally results in decreasing errors (except for $ROM(11)$ and $ROM(21)$ in $L^2(0,T;H^1)$ norm, in which the errors slightly increase). When comparing to the error plot in Fig. 4.28 for Example 4.2, a similar behavior as that in Example 4.2 is observed: with respect to errors, the ROMs perform better as the number of snapshots being higher. For the number of snapshots equal or larger than 251, the error curves are nearly overlapping. The discrete $L^2(0,T;L^2)$ error plot shows that for very low r$\leqslant$15, $ROM(m), m \in \{51, 101, 251, 501, 1001\}$ are similar. For r$\geqslant$50, the ROMs are better than the solution of G-FEM, and approaching closely to the solution of SUPG-FEM, but always not being better than it.
Compare to the errors in $L^2(0,T;L^2)$ norm, the plot of discrete $L^2(0,T;H^1)$ error given in the lower panel of Fig. 4.37 shows some different behaviors: for the amount of snapshots equal or higher than 51, the errors decay linearly as the rank of POD modes increases. In addition, the errors for G-FEM are remarkable, which are much larger than the errors for all ROMs and SUPG-FEM. Unlike in the Example 4.2, concerning the errors in discrete $L^2(0,T;H^1)$ norm, even though the differences between ROMs and SUPG-FEM become fewer and fewer with higher r, the SUPG-FEM always shows better performance.

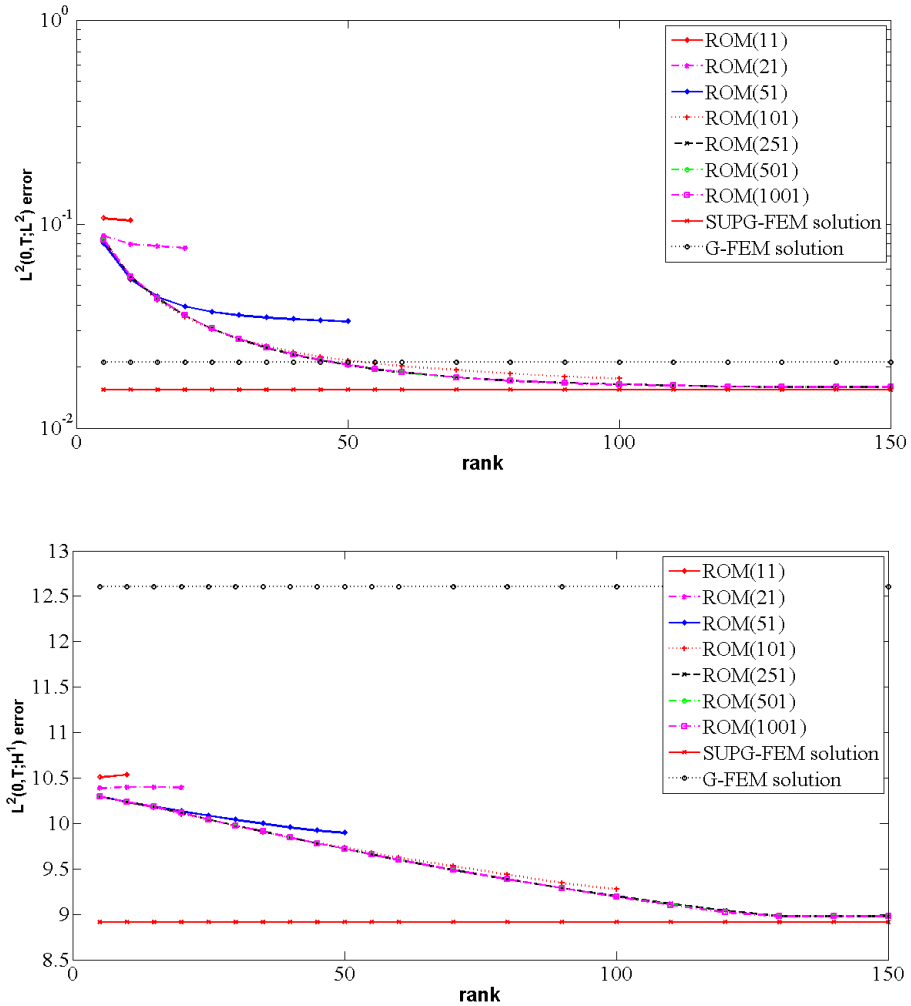


Figure 4.37: Example 4.3.1: $L^2(0,T;L^2(\Omega))$ (top) and $L^2(0,T;H^1(\Omega))$ (bottom) error for different snapshots.

It is worth mentioning that, in Fig. 4.28 for Example 4.2, somehow the errors for $ROM(11)$ and $ROM(21)$ slightly increase in discrete $L^2(0,T;L^2)$ norm and fast increase in discrete $L^2(0,T;H^1)$ norm. For $ROM(51)$, the errors, especially in discrete $L^2(0,T;H^1)$ norm, first decrease and then turn to increase around $r = 30$ with r increasing. Whereas in Fig. 4.37 for this example, the errors for $ROM(11)$, $ROM(21)$ and $ROM(51)$ in both norms gradually decrease or only slightly increase as r increase.
The corresponding temporal evolution of the error in $L^2(\Omega)$ norm for $r = 10, 45, 100, 150$ are depicted in Fig. 4.38. Despite the fact that the error curves are quite oscillating all the time for all r, the trend of error proceeding becomes more stable (the turnings of the error curves are more predictable) with higher r. Moreover, the behaviors in the temporal evolution of errors, which coincide with the averaged errors in Fig.4.37 are observed: $ROMs(51)$ generally behave less accurate, and $ROM(m)s, m \in \{101, 251, 501, 1001\}$ yield similar results for higher r.

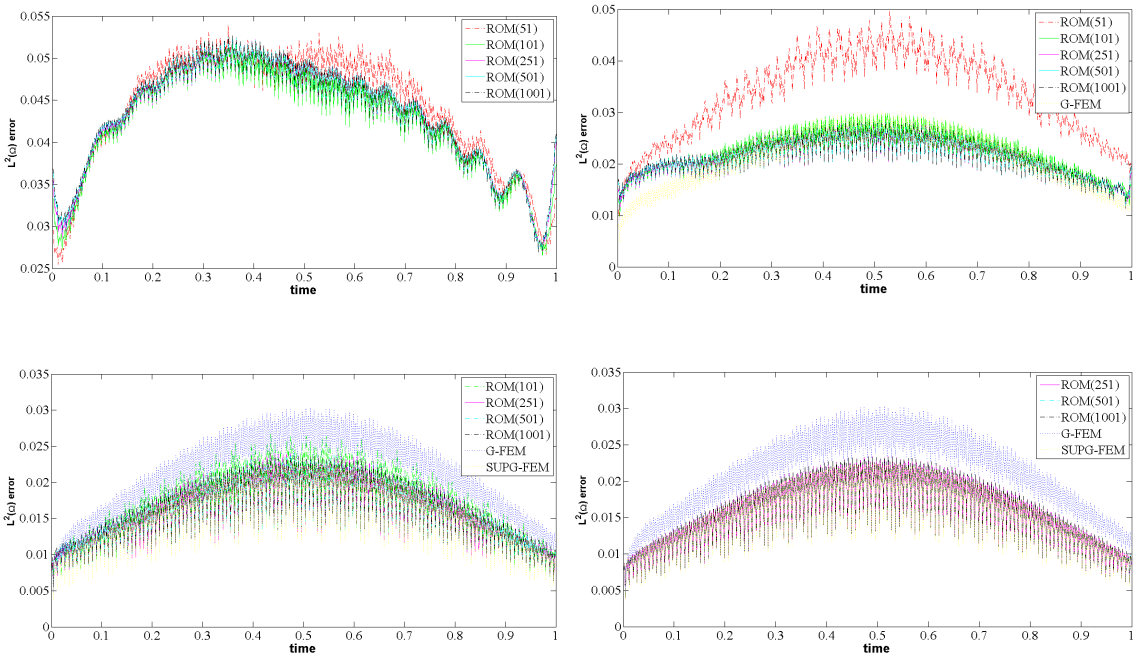


Figure 4.38: Example 4.3.1: Temporal evolution of errors in $L^2(\Omega)$ norm for r=15, 45, 100 and 150 (from top left to bottom right).

Example 4.3.2. *Comparisons of applying different inner products H^1 and L^2 on the computation of the POD modes and the ROMs.*

This example will mainly focus on the following studies: the comparisons of eigenvalues behavior and the corresponding POD modes, of the ROMs, and of the error behaviors between the usage of H^1 and L^2 inner products.
In Example 4.1, the sensitivity of ROMs and the POD modes with respect to different inner products has been studied, and the numerical investigations show that using different inner products almost has same impacts on the results. Since Example 4.1 is a special case, it is also interesting to see whether the use of different inner products will cause different results for Example 4.3.1. For purpose of better comparison, the boundary value problem, the way to calculate and choose the snapshots are chosen to be same as in Example 4.3.1, and the application of inner product H^1 instead of L^2 is the only difference.
Tab. 4.3 shows the rank of the POD modes for each snapshot space with respect to inner product H^1. Similar to the situations in Tab. 4.2 for L^2 inner product, there exists no threshold rank, and the dimension of the snapshot space increases as the number of snapshots grows. In addition, from the plot of corresponding eigenvalues, a steep drop is also observed near r=130. However, the dimension of each of 11, 21, 51, 101, 251 and 501 snapshot spaces reaches its maxima (i.e., one less than the number of snapshots, as the initial value is always preserved), and for 1001 snapshot space, the dimension is even twice larger than the corresponding one in Tab. 4.2. Note that the extra dimentions won't be caused by oscillations due to the usage of the same snapshots, and the usage of H^1 inner product seems to detect more information in the noise. These observations imply that using H^1 inner product yield a more complicated reduced order problem than L^2 inner product. Fig. 4.39 gives the corresponding plot of eigenvalues for each snapshot space. Similarly to the situation in Fig. 4.30 for inner product L^2, the eigenvalues also reduce rapidly around r≈130, which implies that both inner products find the same number of essential modes, $r \approx 130$.

Number of snapshots	11	21	51	101	251	501	1001
Rank of POD modes	10	20	50	100	250	500	872

Table 4.3: Example 4.3.2: Ranks of POD modes for 11, 21, 51, 101, 251, 501 and 1001 snapshots (from left to right) with respect to inner product H^1.

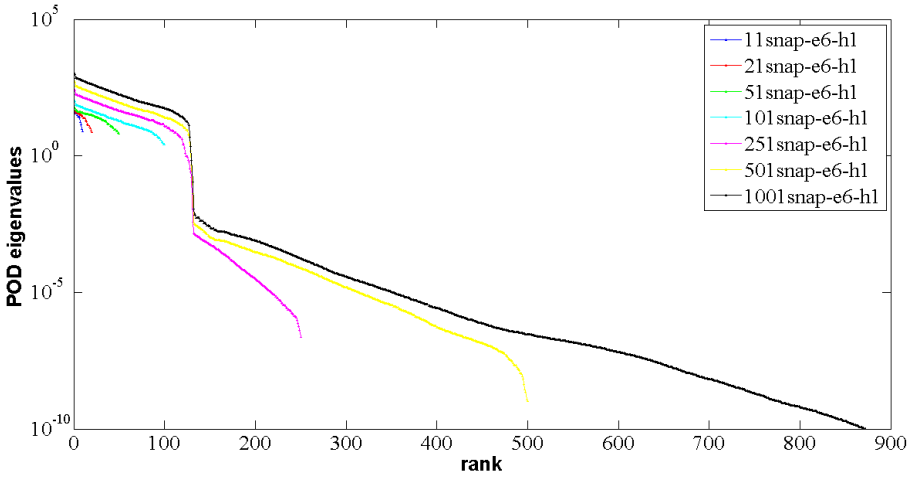


Figure 4.39: Example 4.3.2: POD eigenvalues for 11, 21, 51, 101, 251, 501 and 1001 snapshots with inner product H^1 .

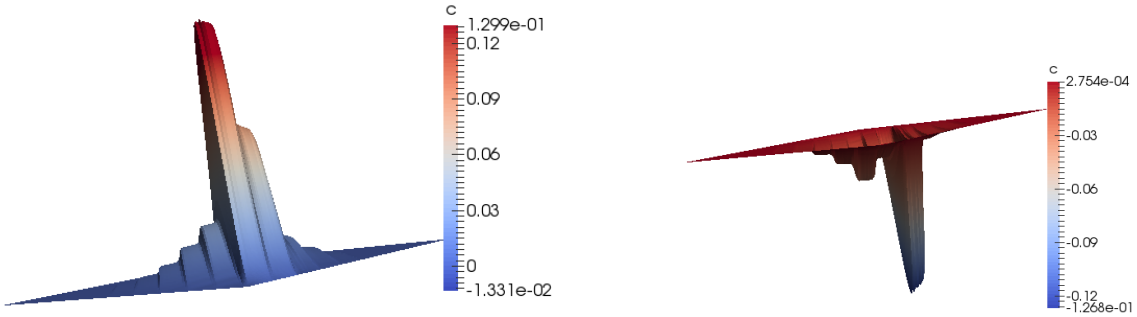


Figure 4.40: Example 4.3.2: $\mathcal{B}^1_{H^1}(11)$ (left) and $\mathcal{B}^2_{H^1}(11)$ (right).

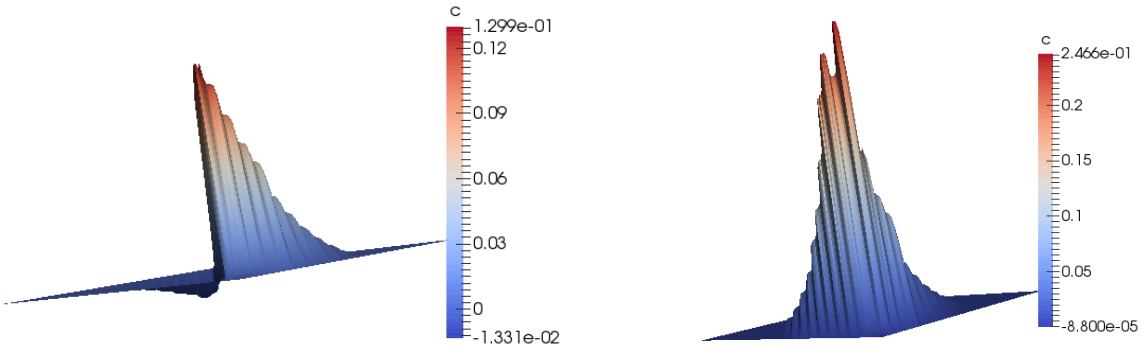


Figure 4.41: Example 4.3.2: $\mathcal{B}^1_{H^1}(21)$ (left) and $\mathcal{B}^2_{H^1}(21)$ (right).

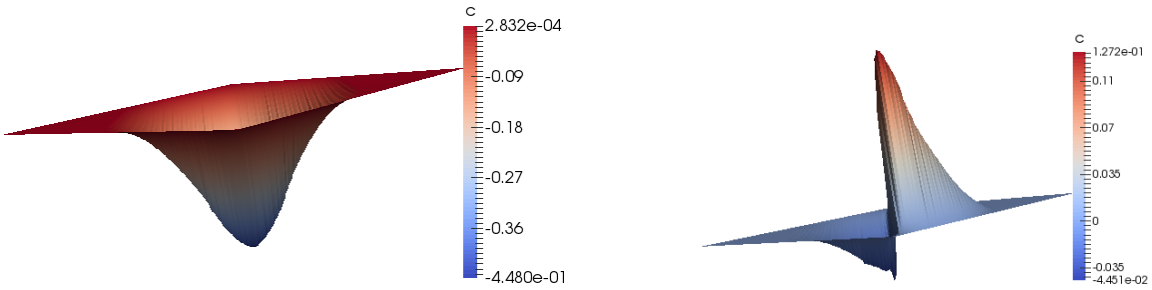


Figure 4.42: Example 4.3.2: $\mathcal{B}^1_{H^1}(51)$ (left) and $\mathcal{B}^2_{H^1}(51)$ (right).

To distinguish B^1 from L^2 inner product, the first POD modes from H^1 inner product are denoted as $B^1_{H^1}$ with the subscript $\{H^1\}$, and the same for ROM_{H^1} and other POD modes. For 11, 21, and 51 snapshots, the first POD modes $B^1_{H^1}(m)$, $m \in \{11, 21, 51\}$ possess distinct forms, see in Figs. 4.40-4.42. This is clearly in contrast to the case for L^2 inner product, since in Example 4.3.1 even though the corresponding POD modes behave different in terms of smoothness for different snapshot spaces, the general shape of the POD modes almost stays the same.

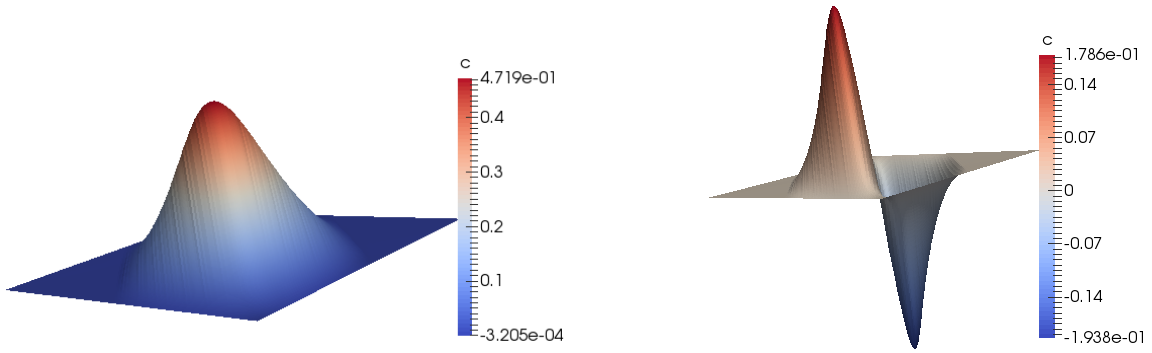


Figure 4.43: Example 4.3.2: $\mathcal{B}^1_{H^1}(101)$ (left) and $\mathcal{B}^2_{H^1}(101)$ (right).

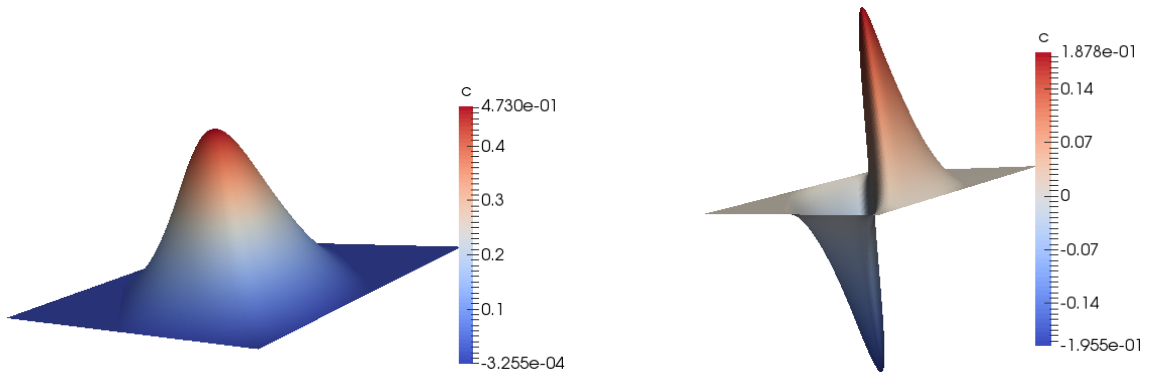


Figure 4.44: Example 4.3.2: $\mathcal{B}^1_{H^1}(251)$ (left) and $\mathcal{B}^2_{H^1}(251)$ (right).

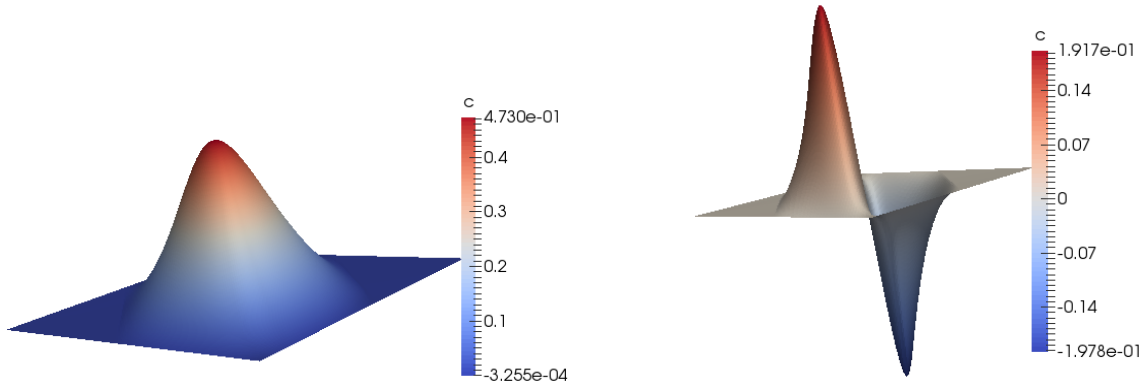


Figure 4.45: Example 4.3.2: $\mathcal{B}^1_{H^1}(501)$ (left) and $\mathcal{B}^2_{H^1}(501)$ (right).

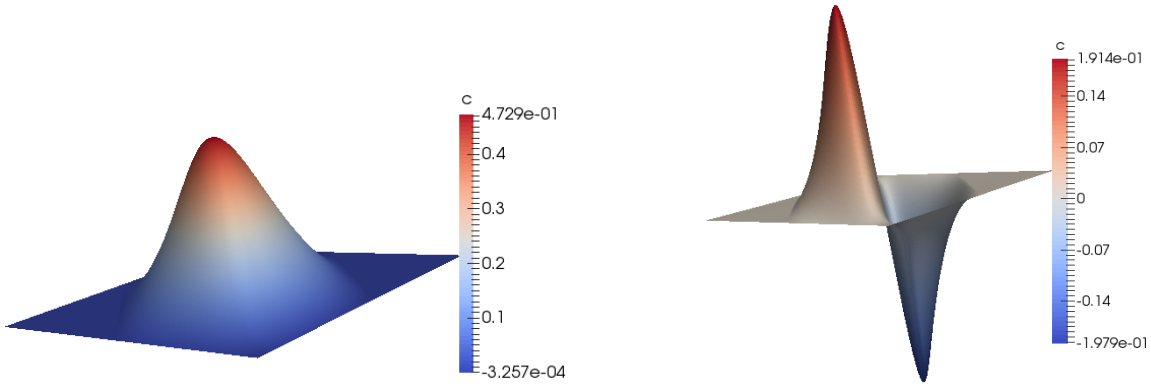


Figure 4.46: Example 4.3.2: $\mathcal{B}^1_{H^1}(1001)$ (left) and $\mathcal{B}^2_{H^1}(1001)$ (right).

Figs. 4.43-4.46 show the first two POD modes for 101, 251, 501 and 1001 snapshot spaces. A similar behavior as that in Example 4.3.1 is observed that the POD modes share almost the same shape as for different snapshot spaces. The pictures of the modes become smoother and smoother as the number of snapshots increases, and the peak of smoothness is firstly reached for 251 snapshots, since for the number of snapshots more than 251, one can barely see any difference.

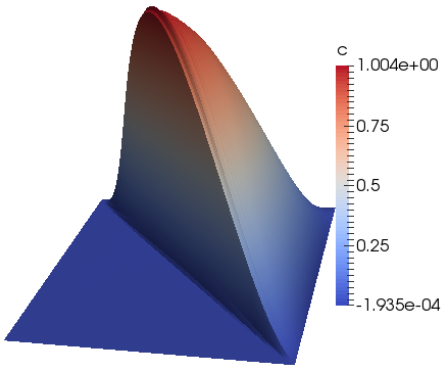


Figure 4.47: Example 4.3.2: SUPG-FEM solution at t=0.5.

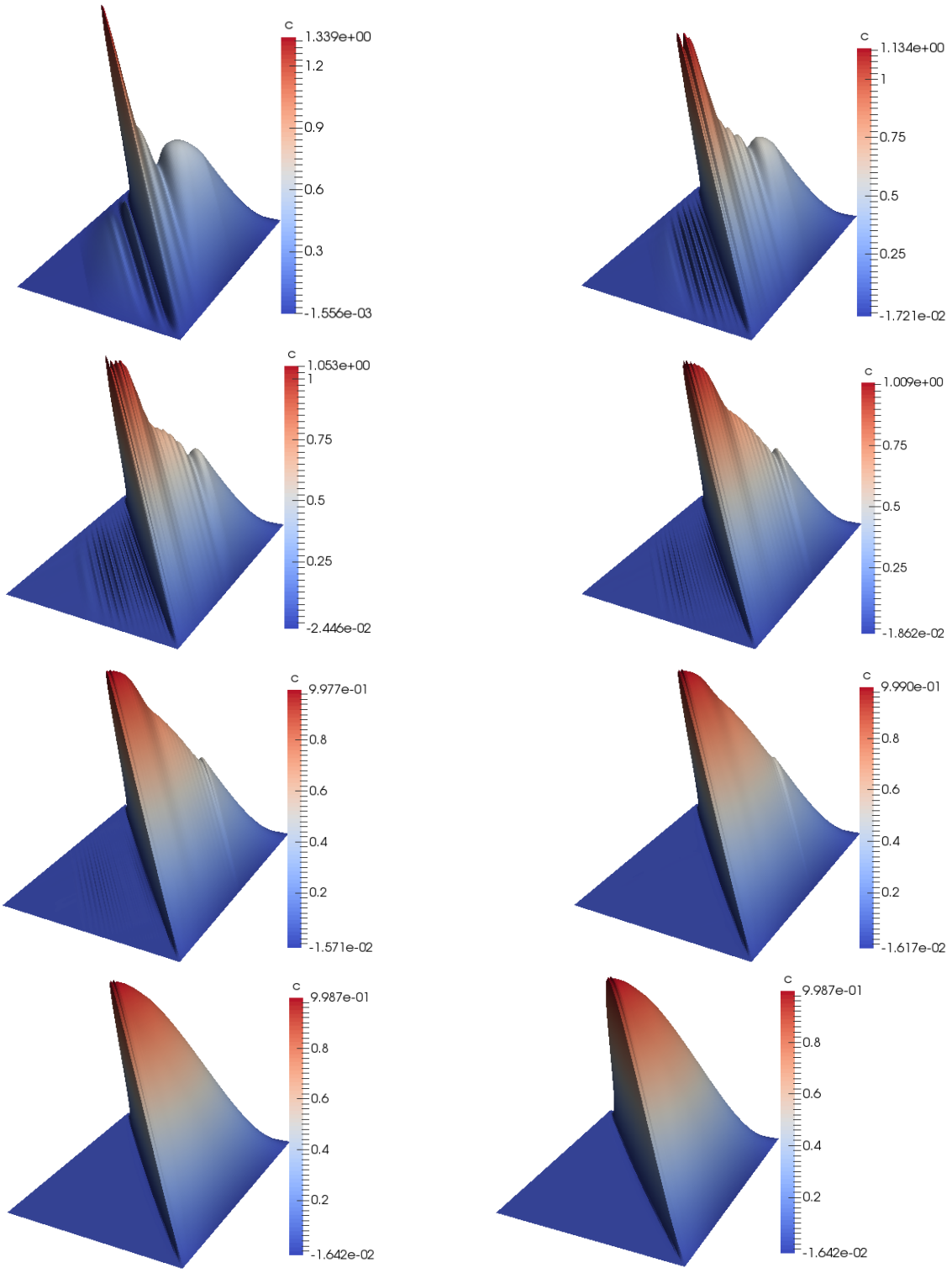


Figure 4.48: Example 4.3.2: $ROM_{H^1}(1001)$ at t=0.5 for r=10, 30, 50, 70, 100, 120, 130, 150 (from top left to bottom right).

For the sake of investigating how the ROM_{H^1} are influenced by r, the $ROM_{H^1}(1001)$ for different r at t=0.5 are represented in Fig. 4.48. Similar to the results explored in Example 4.3.1, as r increases, the spurious oscillations are reduced and the ROMs are

getting closer to the solution of SUPG-FEM shown in Fig. 4.47.

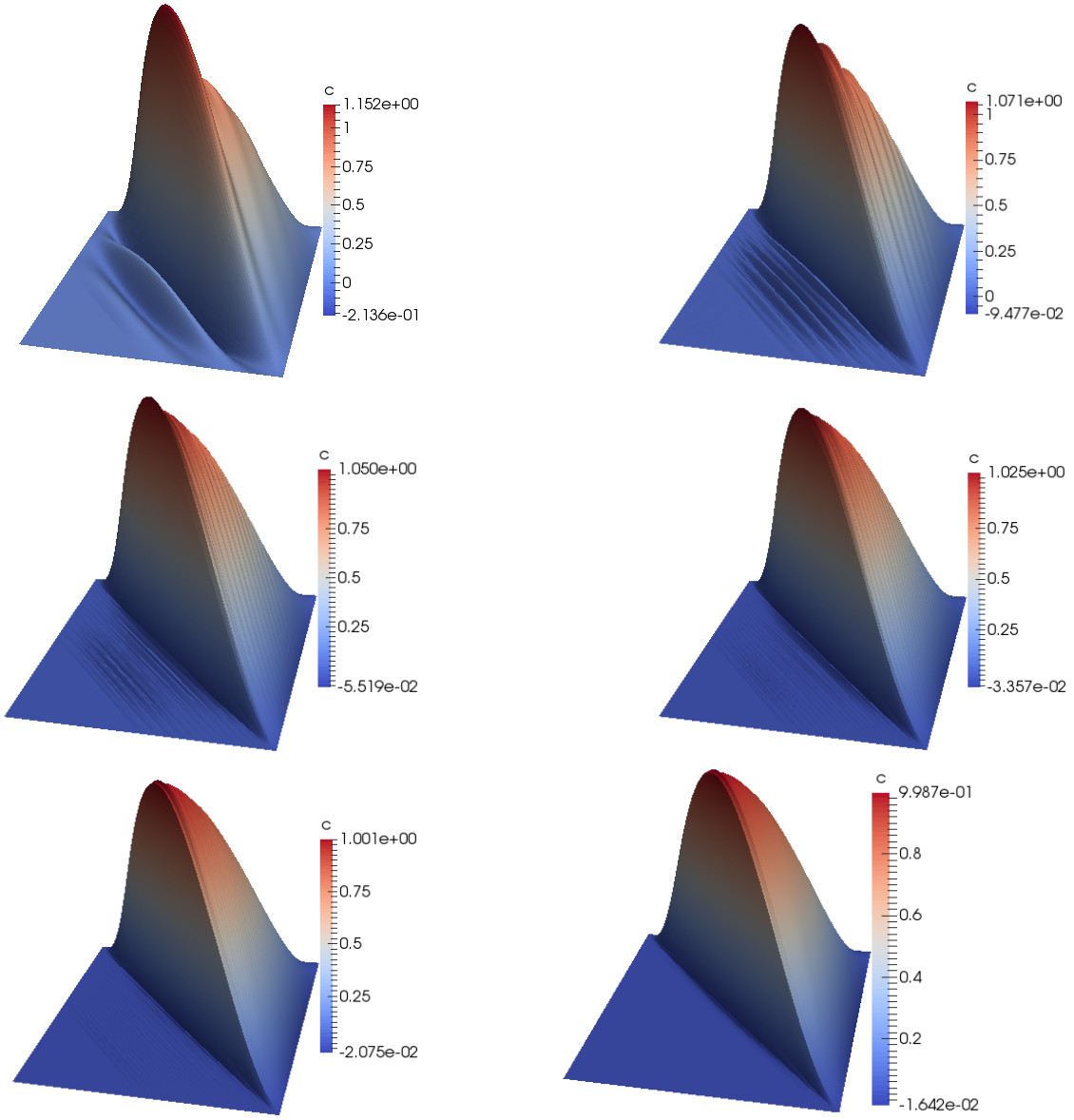


Figure 4.49: Example 4.3.2: $ROM_{L^2}(1001)$ at t=0.5 for r=10, 30, 50, 70, 100, 120 (from top left to bottom right).

For better comparisons of the effects of different inner products on the resulting ROMs, the corresponding ROMs for L^2 inner product at t=0.5 are shown in Fig. 4.49. Even if ROM_{H^1} and yield nearly as accurate results as ROM_{L^2} for r high enough, the ROM_{L^2} still perform better, since for ROM_{H^1} there are always deformations on the cambered surface unless high r is considered.

In Figs. 4.50-4.51, the plots of comparisons of the discrete $L^2(0,T;L^2)$ and $L^2(0,T;H^1)$ errors between $ROM_{H^1}(m)$ and $ROM_{L^2}(m)$, $m \in \{11, 51, 101, 251, 1001\}$ are presented respectively. It is observed that, for both kinds of errors, the errors for ROM_{H^1} are generally larger than the ones for ROM_{L^2}. For each snapshot space, these errors become

consistent for r being close or equal to its maxima.
When comparing to the plots of $L^2(0,T;L^2)$ error in Fig. 4.50, the plots of $L^2(0,T;H^1)$ error in Fig. 4.51 shows some differences: concerning the errors, $ROM_{H^1}(11)$ and $ROM_{H^1}(51)$ for r being equal to its highest rank, i.e., r=10 and 50 respectively, always perform worse than the corresponding ones from snapshot spaces of larger amount. In addition, the differences of errors between $ROM_{L^2}(m)$ and $ROM_{H^1}(m)$,$m \in \{101, 251, 501, 1001\}$, keep decreasing as r increases, and become negligible for $r \approx 70$, which is $r \approx 130$ in the $L^2(0,T;L^2)$ plot. This suggests that, with respect to $L^2(0,T;H^1)$ error, ROM_{L^2} and ROM_{H^1} show similar performances for $r \geqslant 70$.

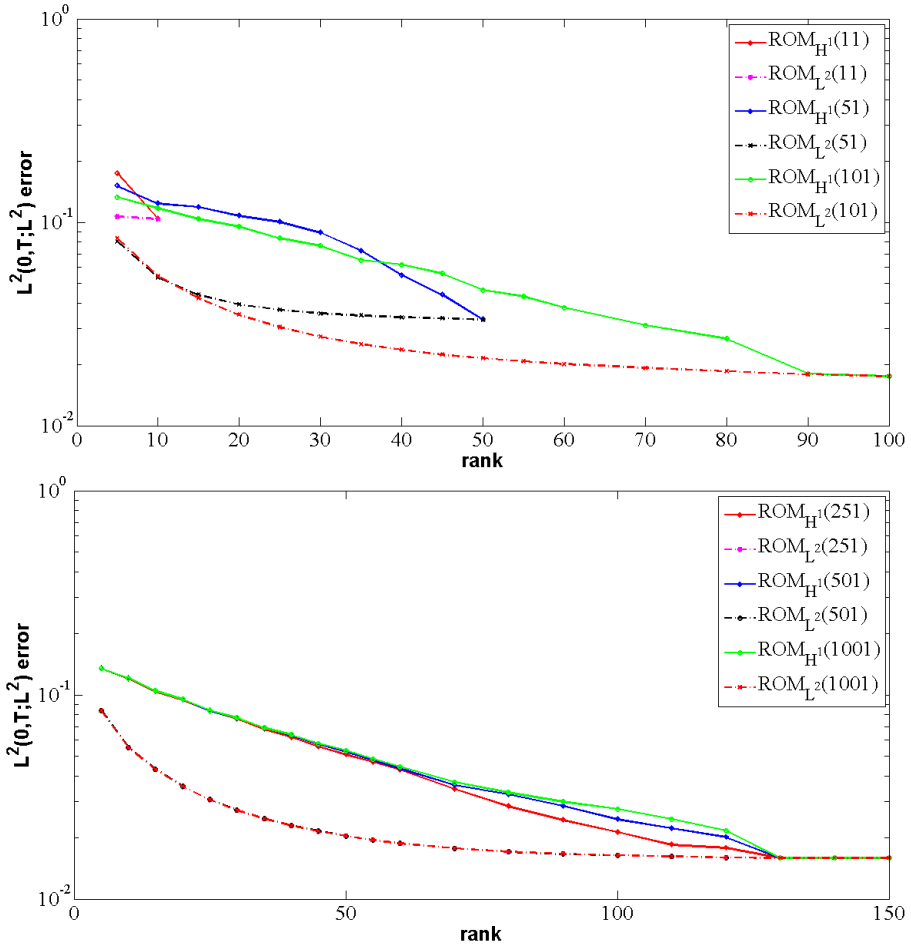


Figure 4.50: Example 4.3.2: Comparisons of discrete $L^2(0,T;L^2)$ errors by ROM_{H^1} and ROM_{L^2} for 11, 51, 101, 251, 501 and 1001 snapshots.

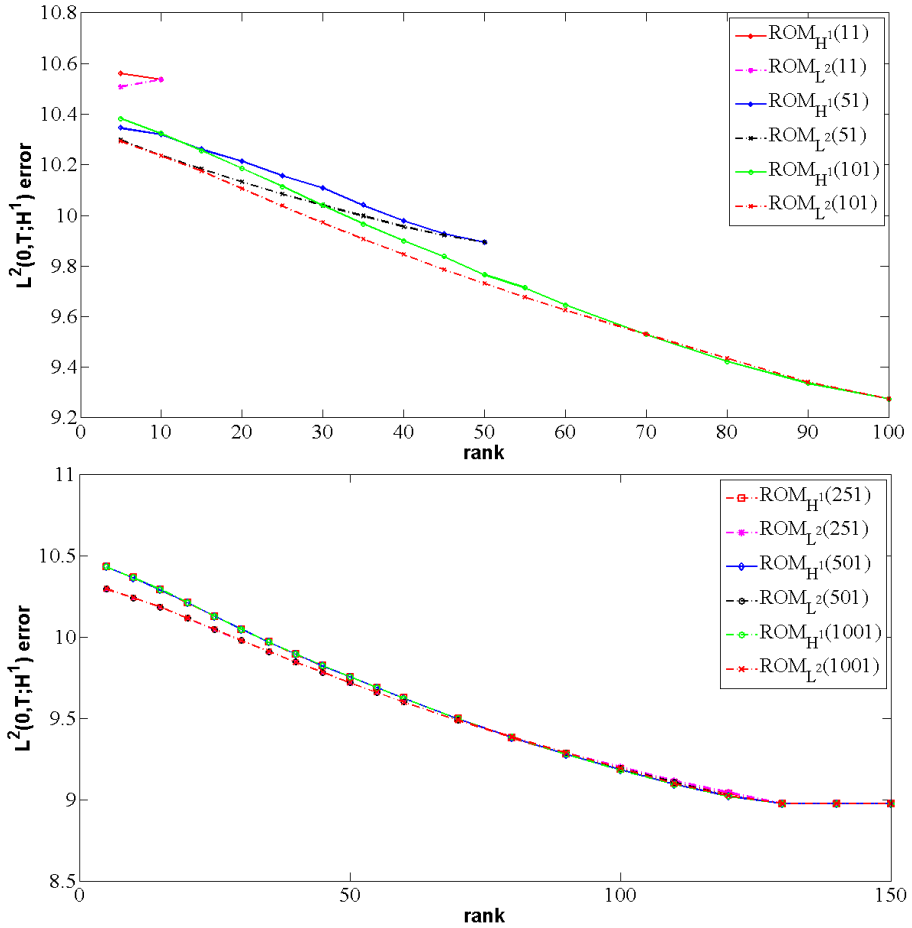


Figure 4.51: Example 4.3.2: Comparisons of discrete $L^2(0,T;H^1)$ errors by ROM_{H^1} and ROM_{H^1} for 11, 51, 101, 251, 501 and 1001 snapshots.

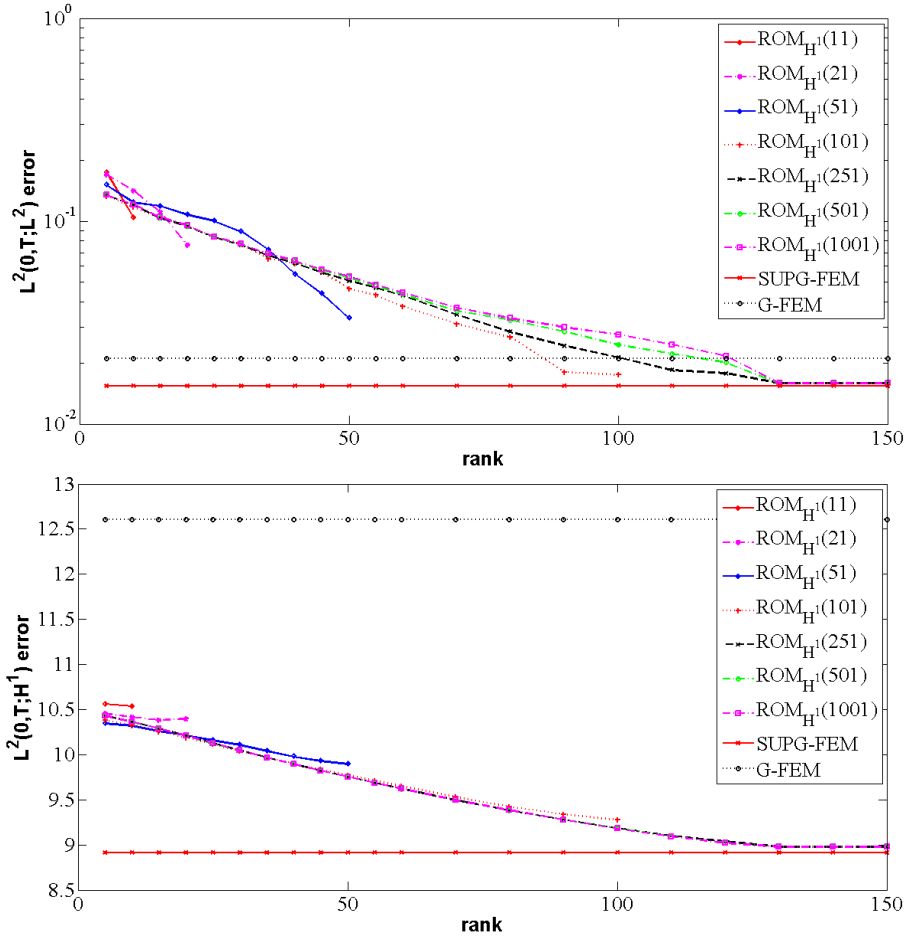


Figure 4.52: Example 4.3.2: $L^2(0,T;L^2)$ (upper) and $L^2(0,T;H^1)$ (bottom) errors for different snapshots.

Fig. 4.52 shows the errors in discrete $L^2(0,T;L^2)$ and $L^2(0,T;H^1)$ norm for ROM_{H^1} and also the solution of SUPG-FEM and G-FEM. The results for errors in $L^2(0,T;H^1)$ norm are very similar to corresponding one in Example. 4.3.1, see Fig. 4.37. Meanwhile, the errors in $L^2(0,T;L^2)$ norm show different behaviors: On one hand, the error curves are quite roughly, which are different from the smooth ones in Fig. 4.37. On the other hand, the error curves for different snapshots intersects at some points. This means that the conclusion in Example. 4.3.1 that the ROMs for larger snapshots generally cause less (or equal) errors does not apply in this example, and implies that the dacay of the error $L^2(0,T;L^2)$ is more unpredictable than that in Example 4.3.1.

Example 4.3.3. *Fetching snapshots from the analytical solution*

In this example, the snapshots were taken directly from the analytical solution, and the POD modes were extracted from these snapshots. The L^2 inner product is applied for the computation of the POD modes and ROMs. As for the study of ROMs, not only the SUPG-ROMs but also the G-ROM were considered, and the comparisons of results from both were given. Last but not least, the comparisons of ROMs and POD modes from interpolations of analytical solution and SUPG-FEM were emphasized.

As a usual routine, the ranks of eigenvalues for each snapshot space is given in Tab. 4.4, and the corresponding eigenvalues are shown in Fig. 4.53. It is observed that, for 251, 501 and 1001 snapshot spaces, there exists a maximal rank $r = 129$, and this maximal rank is much less than the corresponding ones in Tab. 4.2. Remember that in Example 4.3.1, a fast drop of eigenvalues was observed around $r \approx 130$. This phenomenon could be explained by the fact that the POD modes, which correspond to the eigenvalues smaller than the eigenvalues at the dropping point, obtain the information from the spurious oscillations. These POD modes were computed from the 'noisy data', which are inevitable even though the stabilization method was used.

Number of snapshots	11	21	51	101	251	501	1001
Rank of POD modes	10	20	50	100	129	129	129

Table 4.4: Example 4.3.3: Ranks of POD modes for 11, 21, 51, 101, 251, 501 and 1001 snapshots (from left to right).

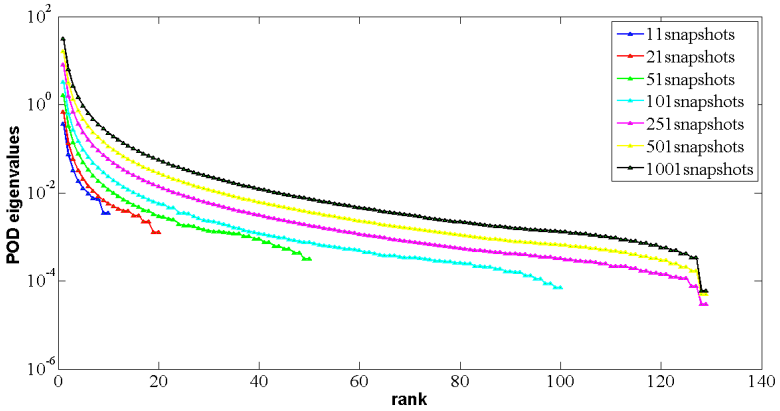


Figure 4.53: Example 4.3.3: POD eigenvalues for 11, 21, 51, 101, 251, 501 and 1001 snapshots.

Firstly, the SUPG-ROMs were built by using the POD modes extracted from snapshots of analytical solution. The plots below show the resulting errors in discrete $L^2(0,T;L^2)$ and $L^2(0,T;H^1)$ norm for each snapshot space.

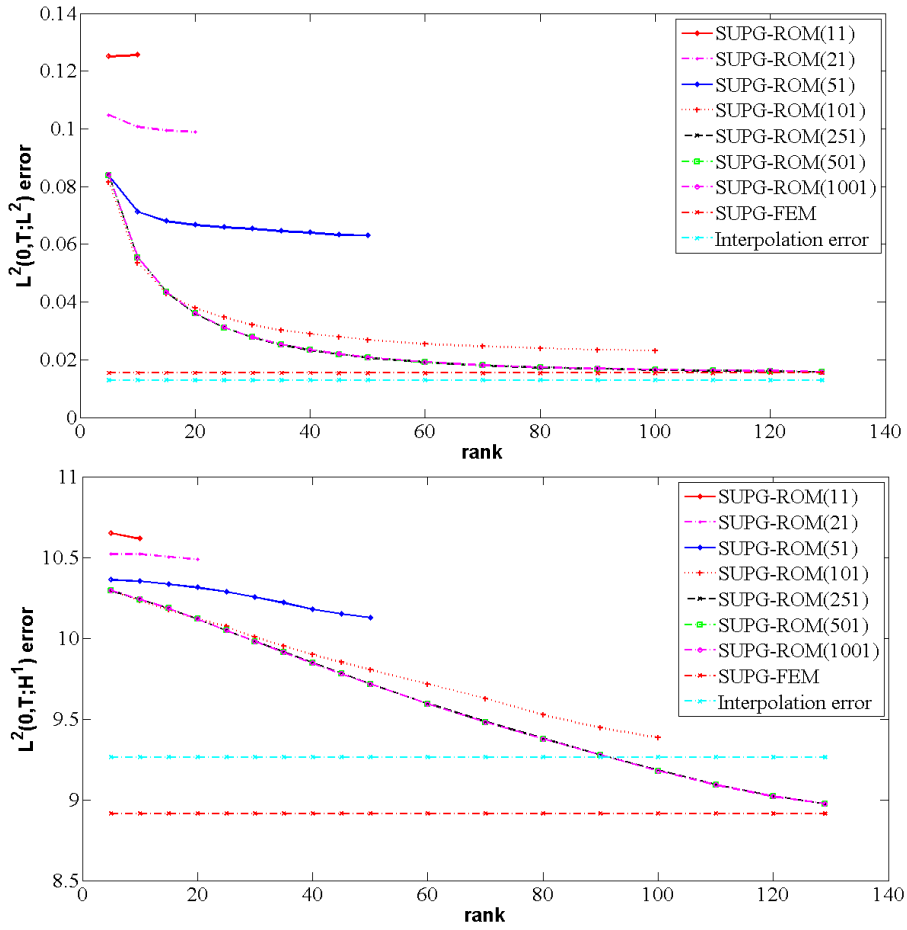


Figure 4.54: Example 4.3.3: $L^2(0,T;L^2(\Omega))$ (top) and $L^2(0,T;H^1(\Omega))$ (bottom) errors for $ROM(m), m \in \{11,\ldots,1001\}$.

Despite the different scalar values, a similar qualitative behavior for both of $L^2(0,T;L^2)$ and $L^2(0,T;H^1)$ errors as that in Fig. 4.37 can be observed in Fig. 4.54: as r increases, the errors decrease (or stay the same for $ROM(11)$). For a fixed r, the ROMs with larger amount of snapshots show better performance with respect to the errors. Furthermore, as both of r and amount of snapshots become large enough, the errors for ROMs closely approach to the SUPG-FEM solution. It is worth noting that, in the plot of $L^2(0,T;L^2)$ error in Fig. 4.54, the errors from interpolations of analytical solution are slightly smaller than the ones from solutions of SUPG-FEM, whereas in the plot of $L^2(0,T;H^1)$ error, the errors from interpolations of analytical solution is bigger than the ones not only from SUPG-FEM solutions but also from SUPG-ROMs for $m \geqslant 251$ and $r \geqslant 90$.

Secondly, the G-ROMs were built by using the same POD modes as for the SUPG-ROMs. A comparison of the errors in discrete $L^2(0,T;L^2)$ norm for these two types of ROMs is given below:

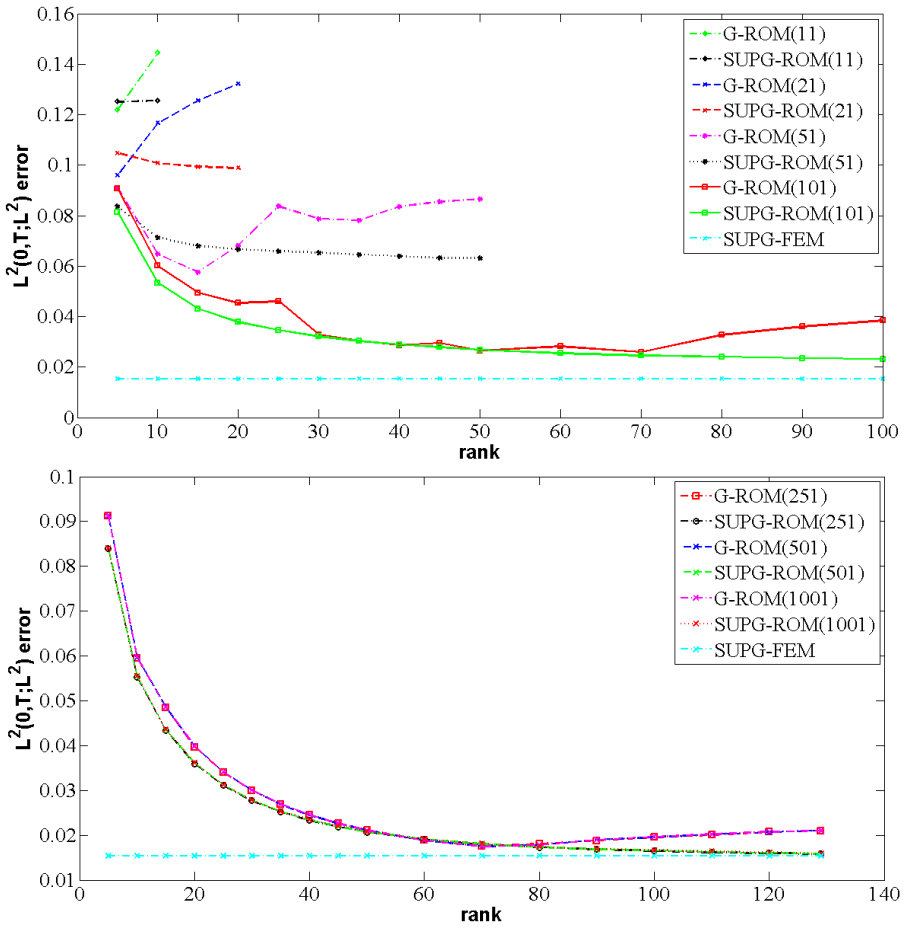


Figure 4.55: Example 4.3.3: Comparisons of errors in $L^2(0,T;L^2(\Omega))$ norms between $G-ROM(m)$ and $SUPG-ROM(m)$, $m \in \{11,\ldots,1001\}$.

The error curves in the top plot of Fig. 4.55 seem quite roughly and erratic. The explaination for this behavior might be that, concerning the errors, the ROMs for lower amount of snapshots (i.e., 11,21,51 and 101) are strongly influenced by the spurious oscillations which occur in the G-ROMs. However, the error plot below for higher amount of snapshots (i.e., 251, 501 and 1001) shows very different results: for $r \leqslant 80$, the errors of G-ROMs progressively reduce with increasing r. The same as that in Example 4.3.1, the errors between the ROMs for different snapshots stay the same, and are close to the errors of SUPG-ROMs. For $r \geqslant 80$, the errors of G-ROMs somehow gradually increase as r increases.

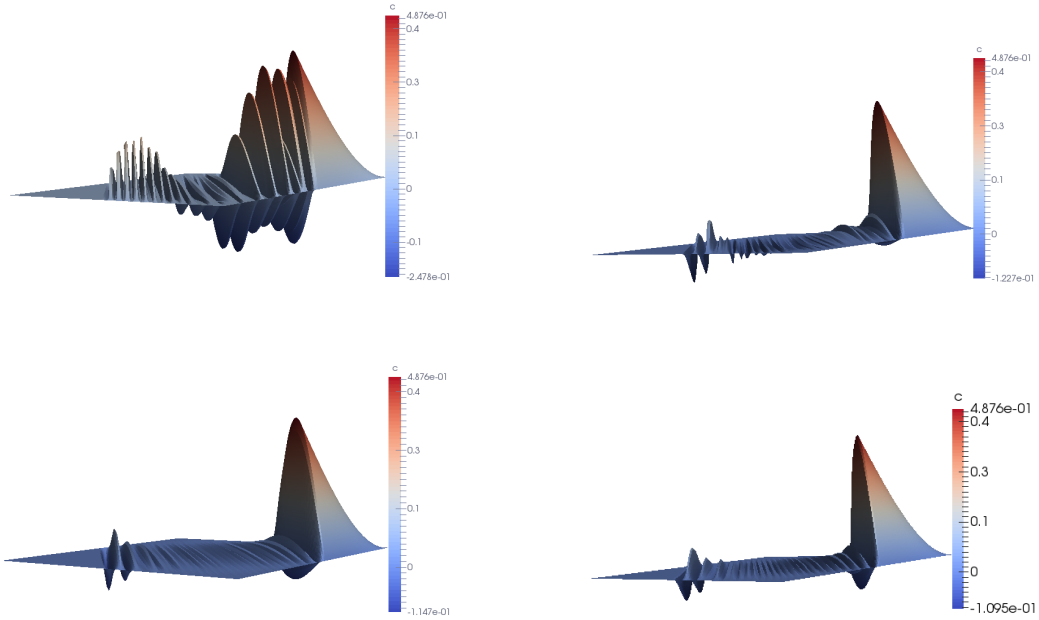


Figure 4.56: Example. 4.3.3: $G - ROMs$ for 51, 101, 251 and 1001 snapshots (from top left to bottom right) at t=1.0 for r=50.

In Fig. 4.56, the $G - ROM(m)$, $m \in \{51, 101, 251, 501\}$, are presented for r=50. The results of these G-ROMs seem to be in conformity with the observation of the error behavior in Fig. 4.55: concerning the oscillations, the G-ROMs for 101, 251 and 501 snapshots perform much better than the G-ROM(51), in which there are huge spurious oscillations. Nevertheless, in spite of the fact that the averaged errors of G-ROMs for higher amount of snapshots are slightly less than the ones of corresponding SUPG-ROMs for r=70, the result of G-ROM may still obtain more oscillations than the corresponding SUPG-ROM at certain time, see Fig. 4.57 for the comparison of $G - ROM(1001)$ and $SUPG - ROM(1001)$ for r=70. From the observations of plots of errors and the pictures of G-ROMs and ROMs, it can be concluded that stabilization method is needed even if the perfect snapshots are used.

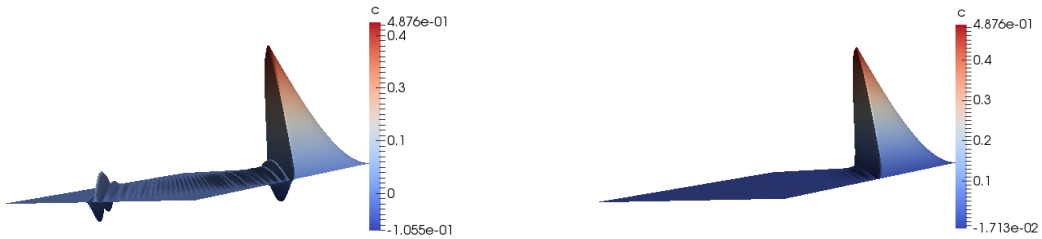


Figure 4.57: Example. 4.3.3: a comparison of $G - ROM(1001)$ and $SUPG - ROM(1001)$ for r=70.

Next step is to make the comparisons between the SUPG-ROMs constructed by POD modes from snapshots of SUPG-FEM solution and from the analytical solution. They will be denoted as $ROM - SUPG(m)$ and $ROM(m)$, $m \in \{11, \ldots, 1001\}$, respectively for short.

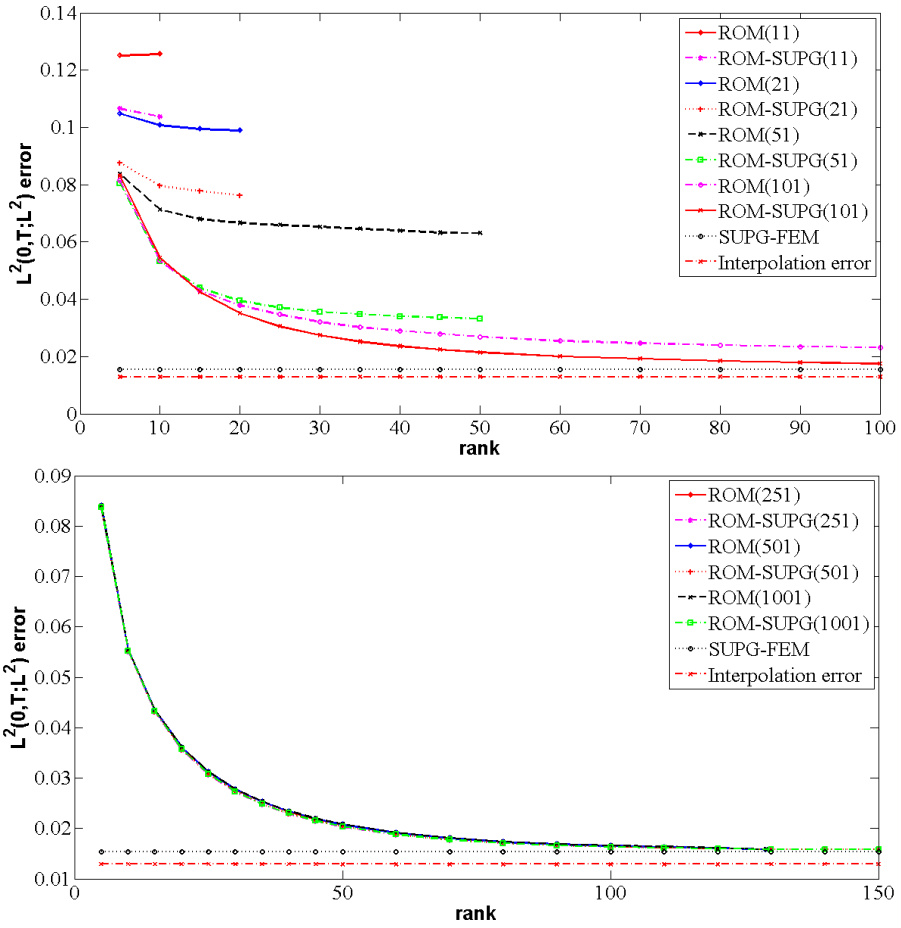


Figure 4.58: Example. 4.3.3: The comparison of discrete $L^2(0,T;L^2)$ errors for $ROM(m)$ and $ROM-SUPG(m)$, $m \in \{11,\ldots,1001\}$.

In Fig. 4.58, the comparison of errors in discrete $L^2(0,T;L^2)$ norm for $ROM(m)$ and $SUPG-ROM(m)$ is shown. As already discussed for the error plot in Fig. 4.54, even though the scalar values of the errors from the ROMs and SUPG-ROMs are different for each snapshot space, the qualitative behaviors are the same. For lower amount of snapshots, i.e., 11, 21, 51 and 101 snapshots, as r increases, the errors from $ROM(m)$ reduce slower and are always larger than the ones from the corresponding $SUPG-ROM(m)$. For higher amount of snapshots, i.e., 251, 501 and 1001 snapshots, the error curves for all the ROMs and ROM-SUPGs are almost overlapping. This implies that once the amount of snapshots is large enough, the ROMs behave nearly the same as the ROM-SUPGs with respect to averaged errors. For the maximal rank $r = 129$ of the POD modes from analytical solution, the ROMs achieve its most accurate result, which seems also the most accurate result for ROM-SUPGs for $r \leqslant 150$. Concerning the errors, as r is high enough, both of the ROMs and ROM-SUPGs closely approximate the solution of SUPG-FEM, but still some way to go when compared with the one from analytical solution.
The corresponding $L^2(0,T;H^1)$ errors are depicted in Fig. 4.59. The similar qualitative behaviors as that for $L^2(0,T;L^2)$ error are observed, except that the errors from interpolations of analytical solution are bigger than then ones from both SUPG-FEM solution

and $ROM(m)$, $SUPG - ROM(m)$ for $m \geqslant 251$ and $r > 90$. Besides, the interpolation errors are almost equal to the ones from $SUPG - ROM(101)$ for r=100.

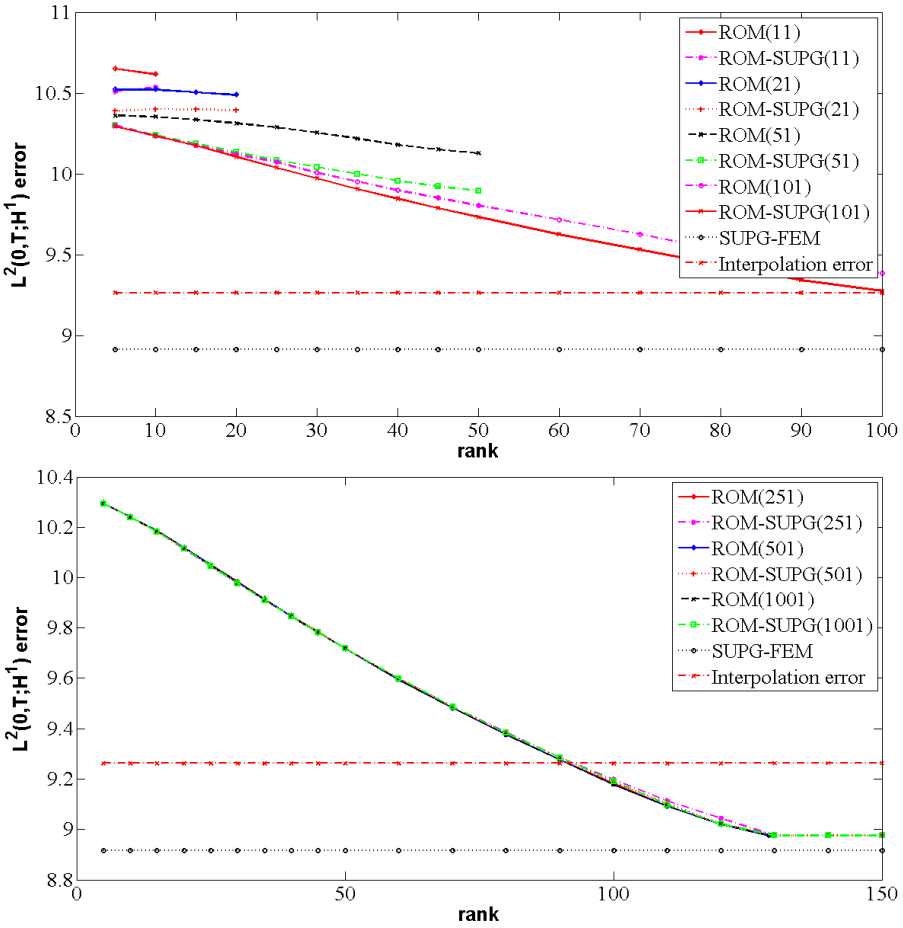
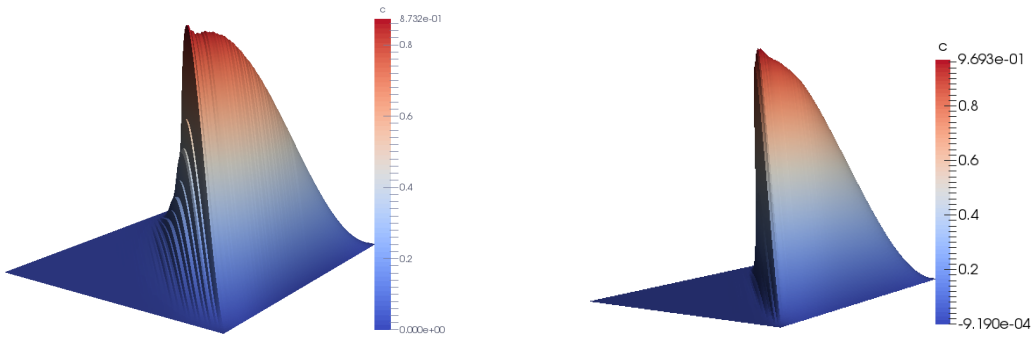


Figure 4.59: Example. 4.3.3: the comparison of discrete $L^2(0,T;H^1)$ errors for $ROM(m)$ and $ROM - SUPG(m)$, $m \in \{11, \ldots, 1001\}$.



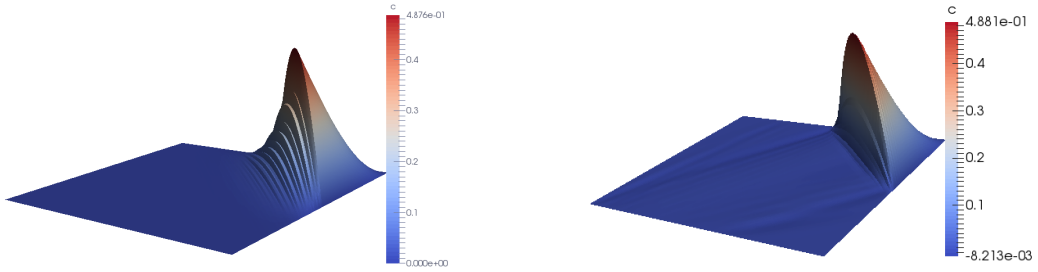


Figure 4.60: Example. 4.3.3: $ROM(51)$ and $ROM - SUPG(51)$(from left to right) for r=50 at t=0.5 and t=1.0(from top to bottom).

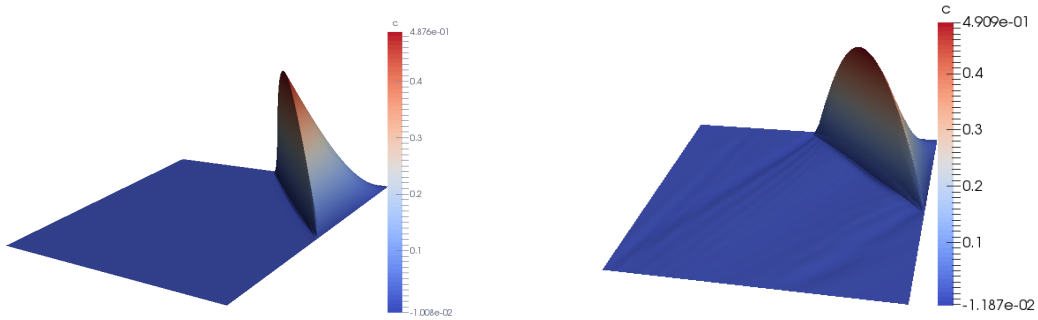


Figure 4.61: Example. 4.3.3: $ROM(1001)$(left) and $ROM - SUPG(1001)$(right) at t=1.0 for r=129,130 respectively.

Fig. 4.60-4.61 give the comparisons of some ROMs and ROM-SUPGs. For the low amount of snapshots, e.g., 51 snapshots, when referring to the values from analytical solution in Fig. 4.62, it is clear that the ROM-SUPG shows more accurate result than the ROM, since the ROMs are smeared especially at the crossing area of the horizontal and perpendicular planes. This is in accordance with the behavior of the averaged errors in Fig. 4.58. However, although both of $ROM(1001)$ and $ROM - SUPG(1001)$ yield good performance, the result of ROM-SUPG at t=1.0 somehow shows that there are slight ripples on the horizontal plane, whereas the plane is quite smooth for $ROM(1001)$. The similar situation also appears for $ROM - SUPG(51)$ (see the bottom right picture in Fig. 4.60), which implies that these ripples come from the SUPG-FEM solution.

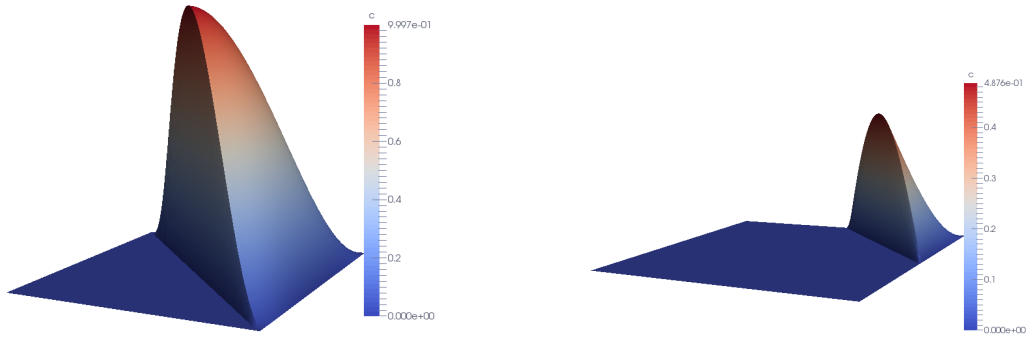


Figure 4.62: Example. 4.3.3: Results taken from analytical solution at t=0.5(left) and t=1.0(right).

5 Summary and Outlook

This thesis presented firstly the derivation of the POD method and the computation of the POD basis functions, and then introduced the convection-dominated convection-diffusion equations on which the POD method was applied. The basic properties and both time and space discretization methods for this type of equation were presented. The SUPG method was used not only for computing the snapshots but also for building the ROMs, due to the fact that the solutions of convetion-dominated problems usually possess sharp layers and cannot successfully resolved by the regular finite element method unless very fine meshes are applied.
The numerical investigations were designed for the purpose of finding the factors which may play a role in the ROMs. This was implemented by studying the sensitivity of ROM simulations with respect to these factors, or making comparisons of the resulting ROMs with adjusting the magnitude or types of these factors. The numerical tests were carried out in the following examples:

- For relatively simple convection-dominated HUMP example, both of G-FEM and SUPG-FEM with three spatial levels of grid were applied for computing the snapshots. L^2, H^1 and SD inner products were used for the POD modes and ROMs.
- For more complex traveling wave problems, firstly, the non-convection-dominated problem was considered, and G-FEM was used for computing the snapshots and building the ROMs. To make better comparisons, afterwards by only changing the constant diffusion coefficients ε, the convection-dominated problem was obtained, and consequently the SUPG method was considered.
- Still the traveling wave problems for convection-dominated case was considered, but using two different inner products of L^2 and H^1 for the computation of POD modes and the construction of ROMs.
- The snapshots were fetched from the analytical solution, and the comparisons of the resulting POD modes and the ROMs with the corresponding ones from SUPG-FEM were made.
- The impact of different amount of snapshots fetched from the finite element solutions on resulting ROMs were considered in all the investigations of traveling wave problems for both non-convection-dominated and convection-dominated case.

The numerical investigations gave the following conclusions:

- For convection-dominated problem, the G-FEM fails to offer a reasonable solution, and the use of stabilization finite element method is necessary. As the grids become finer, both finite element solution and the ROMs become more accurate.
- The usage of different inner products seem yield very similar results for the simple problems. However, for more complex problems, by observing the POD modes and the resulting ROMs, one finds some differences. This implies that, for the goal of getting the wanted accuracy of the solution with least computations, it is helpful to use the suitable inner product for different problems.

- The SUPG method can yield good results, which may obtain only small differences from the ones by analytical solution. Yet there are always small spurious oscillations, and these numerical artefects will be inherited in the ROMs.
- Choosing better snapshots doesn't guarantee the better ROMs, the good choice of the finite element method for the construction of the ROMs also plays an important role.

The investigations in this thesis are mainly heuristic, and the conclusions from the observations strongly depend on the specified problem for the lack of theoretical support. Some questions are left to be answered, such as the irregular behavior of errors for certain inner product and the G-ROMs, the reason why better snapshots don't yield better results, etc,. Hence further investigations and studies are needed to fully understand these behaviors and solve the open questions.

Bibliography

[1] M. Bergmann, C.-H. Bruneau, and A. Iollo. Enablers for robust {POD} models. *Journal of Computational Physics*, 228(2):516 – 538, 2009.

[2] Alexander N. Brooks and Thomas J.R. Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(13):199 – 259, 1982.

[3] Alfonso Caiazzo, Traian Iliescu, Volker John, and Swetlana Schyschlowa. A numerical investigation of velocity-pressure reduced order models for incompressible flows. *J. Comput. Phys.*, 259:598–616, 2014.

[4] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, April 2000.

[5] Swetlana Giere, Traian Iliescu, Volker John, and David Wells. {SUPG} reduced order models for convection-dominated convectiondiffusionreaction equations. *Computer Methods in Applied Mechanics and Engineering*, 289:454 – 474, 2015.

[6] Vivette Girault. *Finite element approximation of the Navier-Stokes equations*, volume 749. Berlin [u.a.] : Springer, Berlin [u.a.], 1979.

[7] Jean-Luc Guermond. Stabilization of galerkin approximations of transport equations by subgrid modeling. *ESAIM: Mathematical Modelling and Numerical Analysis - Modlisation Mathmatique et Analyse Numrique*, 33(6):1293–1316, 1999.

[8] Volker John. Numerical mathematics iii - partial differential equations. Lecture Notes, Free University Berlin, 2013.

[9] Volker John. Numerical mathematics iv : convection-dominated problems, multigrid methods. Lecture Notes, Free University Berlin, 2014.

[10] Volker John. Numerical methods for incompressible flow problems ii. Lecture Notes, Free University Berlin, 2015.

[11] Volker John and Gunar Matthies. MooNMD—a program package based on mapped finite element methods. *Comput. Vis. Sci.*, 6(2-3):163–169, 2004.

[12] Volker John and Julia Novo. Error analysis of the supg finite element discretization of evolutionary convection-diffusion-reaction equations. *SIAM J. Numerical Analysis*, 49(3):1149–1176, 2011.

[13] Volker John and Ellen Schmeyer. Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion. *Comput. Methods Appl. Mech. Engrg.*, 198(3-4):475–494, 2008.

[14] Volker John and Ellen Schmeyer. Finite element methods for time-dependent convectiondiffusionreaction equations with small diffusion. *Computer Methods in Applied Mechanics and Engineering*, 198(34):475 – 494, 2008.

[15] V. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176, 1980.

[16] Y.C. LIANG, H.P. LEE, S.P. LIM, W.Z. LIN, K.H. LEE, and C.G. WU. {PROPER} {ORTHOGONAL} {DECOMPOSITION} {AND} {ITS} application-spart i: {THEORY}. *Journal of Sound and Vibration*, 252(3):527 – 544, 2002.

[17] Robert G. Parr. *Density functional theory of atoms and molecules*, volume 16. New York : Oxford Univ. Press [u.a.], New York, 1989.

[18] Alfio M. Quarteroni and Alberto Valli. *Numerical Approximation of Partial Differential Equations*. Springer Publishing Company, Incorporated, 1st ed. 1994. 2nd printing edition, 2008.

[19] H. Roos, M. Stynes, and L. Tobiska. *Robust numerical methods for singularly perturbed differential equations: convection-diffusion-reaction and flow problems*. Springer series in computational mathematics. Springer, 2008.

[20] L. Sirovich. Turbulence and the dynamics of coherent structures. parts I–III. *Quart. Appl. Math.*, 45(3):561–582, 1987.

[21] Pengtao Sun, Long Chen, and Jinchao Xu. Numerical studies of adaptive finite element methods for two dimensional convection-dominated problems. *J. Sci. Comput.*, 43(1):24–43, 2010.

[22] S. Volkwein. Model reduction using proper orthogonal decomposition. *Lecture Notes, Faculty of Mathematics and Statistics, University of Konstanz*, 2011.